STRUCTURAL COMPLEXITY OF SYSTEM REQUIREMENTS AND ITS IMPLICATIONS FOR THE DEVELOPMENT PROCESS

by

Maximilian Vierlboeck

A DISSERTATION

Submitted to the Faculty of the Stevens Institute of Technology
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Maximilian Vierlboeck, Candidate   Date

ADVISORY COMMITTEE

Dr. Roshanak Nilchiani, Chairman   Date

Dr. Paul Grogan   Date

Dr. Ting Liao   Date

Dr. Sven Esche   Date

STEVENS INSTITUTE OF TECHNOLOGY
Castle Point on Hudson
Hoboken, NJ 07030
2023

STRUCTURAL COMPLEXITY OF SYSTEM REQUIREMENTS AND ITS IMPLICATIONS FOR THE DEVELOPMENT PROCESS

ABSTRACT

Over time, engineered systems have become more and more complex due to ever-increasing performance and property demands. This increase in demands and the resulting complexity causes numerous challenges, resulting in frequent cost and schedule overruns that can be detrimental to system development projects and entire institutions.

In systems, complexity stems from the number of different elements and their respective interactions. One factor that significantly affects these aspects, and the system as a whole, is requirements, which are defined at the beginning of a development process. Due to this prominent position, requirements influence all development process steps. Thus, addressing and measuring complexity from the requirement stage onward is crucial to prevent unwanted behaviors and emergence.

This dissertation presents a novel approach that allows for the analysis of structural complexity based on textual requirements. Requirement structures are generated by eliciting explicit as well as implicit connections from a set of requirements through the application of Natural Language Processing. For these structures, measurement approaches are selected and presented, including spectral theory and information-based metrics that, as a result, enable the quantification of requirement complexity.

The elicitation of requirement structures has been tested in two case studies, including one based on a current research development project of an unmanned aerial vehicle. The results show that high precision (over 99 percent) and low error rates for the definition of requirement structure can be achieved with current Natural Language Processing tools and libraries.

To assess the effect of the quantified structural requirement complexity regarding the development process, a third case study was used to evaluate correlations of the selected metrics and human effort for tridimensional molecular integration tasks measured by time. The results of the third case study indicate that higher requirement complexity measures strongly correlate with higher effort and, by implication, potential development time and cost.

This dissertation shows that structure can be elicited from textual requirement sets and that the complexity thereof can be quantified. Based on this quantification, the discovered correlations show that the structural complexity of requirements can have a measurable effect on the development process.

Author: Maximilian Vierlboeck

Advisor: Dr. Roshanak Nilchiani

Date: April 30, 2023

Department: School of Systems and Enterprises

Degree: Doctor of Philosophy

DEDICATION

To Kimberly, Helga, Roland, and Ludwig.

ACKNOWLEDGEMENTS

I want to thank my wife, Kimberly, for all her support, understanding, and encouragement throughout this journey. I am grateful to my parents, Roland and Helga, for supporting me throughout my life and providing the tools and discipline that enabled me to attempt a dissertation endeavor.

Of course, I would like to express my deepest gratitude to Prof. Roshanak Nilchiani, Roshi, my dissertation advisor, for her guidance, support, and all the knowledge she shared with me through the uncertain times we all faced in the past few years.

I thank my Dissertation Advisory Committee, Prof. Paul Grogan, Prof. Ting Liao, and Prof. Sven Esche, for all the open-minded discussions and invaluable constructive feedback. Their instrumental input helped me produce and significantly improve this dissertation's content. I sincerely thank Dr. Mark Blackburn for letting me work with his team at SERC and for all the input and data he provided to make this dissertation possible.

Lastly, I want to acknowledge those who have influenced my research throughout this journey. Daniel and Raif, thank you for the camaraderie and exciting discussions on and off-topic. Christine and Abhro, thank you for all the amazing research collaborations. Antonio and Alejandro, thank you for your input, help, and honest feedback. Deborah, Mo, Kristyn, Michael, and Dov, I will not only miss our committee meetings. And, of course, to everyone at the Stevens Institute of Technology who has made this one heck of a journey, thank you all!

Table of Contents

LIST OF FIGURES

List of Tables

List of Acronyms and Abbreviations

AI ... Artificial Intelligence
CAD ... Computer Aided Design
CONOPS ... Concept of Operations
COTS ... Commercial Off-the-Shelf
CPM ... Characteristics-Properties Modeling
CSV ... Comma Separated Values
DRM ... Design Research Methodology
ECSS ... European Cooperation for Space Standardization
GE ... Graph Energy
IEC ... International Electrotechnical Commission
IEEE ... Institute of Electrical and Electronics Engineers
INCOSE ... International Council of Systems Engineering
ISO ... International Organization for Standardization
LGE ... Laplacian Graph Energy
MBSE ... Model-Based Systems Engineering
ML ... Machine Learning
MT ... Machine Translation
NASA ... National Aeronautics and Space Administration
NER ... Named Entity Recognition
NLP ... Natural Language Processing
NLP4RE ... Natural Language Processing for Requirements Engineering
NLU ... Natural Language Understanding
NP ... Noun Phrase
PnR ... Precision and Recall

POS .... Parts-of-Speech
RE .... Requirements Engineering
REQ .... Requirement
SE .... Systems Engineering
SERC .... Systems Engineering Research Center
SREM .... Software Requirements Engineering Methodology
SWRL .... Semantic Web Rule Language
TBD .... To Be Defined
TVC .... Thrust Vector Controller
UAV .... Unmanned Aerial Vehicle
VP .... Verb Phrase

CHAPTER 1: INTRODUCTION AND CONTENT

"For progress there is no cure."

John von Neumann

1.1 Introduction

Over time, engineered systems have become more and more complex. A few thousand years ago, humans manufactured simple hardware tools made from a few parts. Today, we have networks that can connect everyone on Earth (and most likely soon beyond), build computer-based artificial intelligence, and design spacecrafts with hundreds of thousands, sometimes even millions of parts (illustrated in Figure 1.1). While this exponential growth and technology improvement is likely to continue, increasing demands regarding performance, functions, and other lifecycle properties create a plethora of problems that affect all steps of the system development process. The result of these problems are significant challenges that cause project time and cost overruns, development stops and stalling, errors, as well as general product failures (Fruhlinger, Wailgum, & Sayer, 2020).

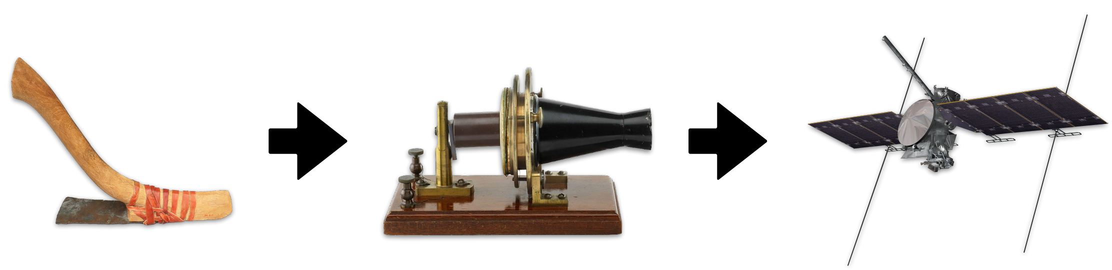

Fig. 1.1 - Technological Progress: Carpenter's Adze, ca. 1479–1458 B.C. (Metropolitan Museum of Art); experimental telephone, 1876 (National Museum of American History); Europa Clipper (National Aeronautics and Space Administration (NASA) & Jet Propulsion Laboratory), Planned to launch in October 2024

Given the extent of the problems described, one might inquire about what causes the issues mentioned earlier to occur. At its core, the challenges and issues stem from the fact that modern engineered systems are characterized by a highly interwoven architecture that is home to a plethora of interactions to fulfill the increasing demands. These characteristics make the system complicated since a single person cannot understand all of the interactions anymore as a whole. Furthermore, if the number of interactions between parts is so large that emergent behaviors, which are impossible to predict, appear, complexity develops and causes unexpected issues/errors, and as a result, cost increases/overruns (Sheard & Mostashari, 2010)(for a precise definition and distinction between complexity and complicatedness, see Chapter 2). Thus, complex systems can be more expensive, less reliable, and prone to failure.

The described issues may make it seem like complexity is inherently negative and has the potential only to make matters worse. This notion is not entirely correct though, since complexity, as long as it is under control, can be positive. A complex system that is controlled can yield a significant competitive advantage that is difficult to compensate for by competitors (Lindemann, Maurer, & Bran, 2009). For instance, if a company manages to control the complexity of a system, it can effectively handle said system, whereas a competitor might struggle with unforeseen emergent behaviors. Furthermore, advanced knowledge and insights regarding complexity can facilitate and propel innovation as avoiding unforeseen complications allows more time for problem-solving and solution generation.

Considering the aspects above, controlling or hedging complexity becomes a worthwhile objective to avoid negative consequences and foster positive ones. However, to control or manage something, one needs to understand it in a way that allows us to measure its effects, changes, and overall dynamics. Thus, it is not surprising that quantifying complexity has been attempted by various scholars and researchers, as shown in Chapter 2 in detail.

Throughout the system development & design process, complexity can be introduced and caused at all steps. Nevertheless, one specific aspect of the process stands out due to its prominent position at the beginning: the defined requirements. To illustrate this position, Figure 1.2 depicts an excerpt from the NASA Systems Engineering (SE) Handbook (National Aeronautics and Space Administration (NASA), 2020), which shows the definition of requirements as the second step of the system development & design process (Figure 1.2).

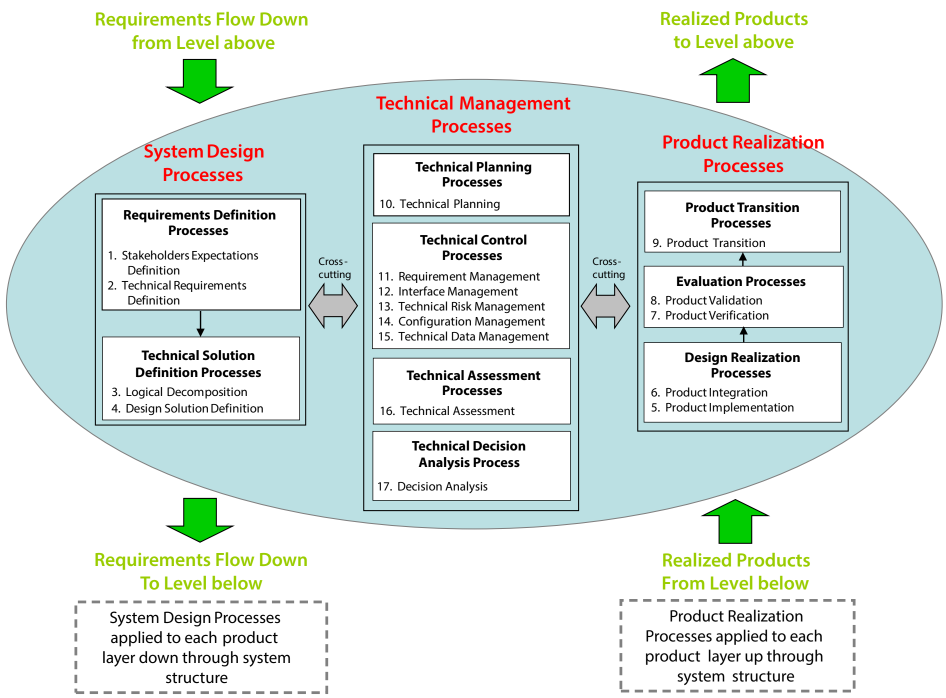

FIG. 1.2 - SE ENGINE (NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA), 2020)

Since requirements are defined at the beginning of the system development process, they influence all steps and decisions that follow. Thus, requirements, directly and indirectly, affect system complexity. In addition, the impact that requirements have is disproportionate, which is further exacerbated by system/design changes becoming more difficult and costly over time with increasing development progress (Boznak, 1994; Boznak & Decker, 1993; Lindemann & Reichwald, 1998; National Aeronautics and Space Administration (NASA), 2020). These consequences are why a management discipline for changes and their implementation exists (Kleedörfer, 1999; Lindemann & Reichwald, 1998; Wildemann, 2020).

Two historical examples (which will surface again in later chapters) help to illustrate the effects that requirements can have: the Douglas DC-1 through DC-3 airplane family ("DC-1 Request for Proposal," 2018) and B-52 aircraft (shown in Figure 1.3).

The design of the DC aircraft family was introduced in 1935, and surprisingly, the requirement specification was written on a single page that contained less than 150 words ("DC-1 Request for Proposal," 2018). This small set of requirements left significant room for design decisions and, as a result, may have contributed to the outstanding quality of the outcome. Similar circumstances apply to the B-52 aircraft, which had fewer than ten pages of requirements. The first B-52 flew in 1952 and, due to various successful modifications over time, is still in use today (as of April 2023) with a planned timeframe until 2045.

Now, one might deduce that fewer requirements are preferable and that they should be kept to a minimum to avoid complexity. While this theory seems intuitive, it cannot be applied without understanding the dynamics. Thus, scientific research regarding connections between requirements and complexity has merit based on the necessity to comprehend the dynamics.

FIG. 1.3 - BOEING B-52 STRATOFORTRESS (2021)

Lastly, the nature and shape of requirements is also important to consider. Traditionally, requirements have been described and documented in text form. While standards exist, as we will see in Chapter 3, the content of requirements and their shape can vary. Hence, measuring and gauging a metric based on requirements must include the analysis of text and language. Since the mentioned text is written and provided as natural language, the machine-assisted approach of Natural Language Processing (NLP) is used for the research in this dissertation to bridge the gap between textual requirements and analysis foundations.

Note that novel approaches exist which represent requirements in model-centric forms (Chammard, Regalia, Karban, & Gomes, 2020; Karban, Dekens, Herzig, Elaasar, & Jankevičius, 2016), but the predominant shape as of the time of this writing (April 2023) are still documents characterized by natural language.

Bringing together the need to measure and or gauge complexity with its connection to requirements in the development process formed the starting point for the research topic of this dissertation in form of the following two questions: 1) how are requirements related to complexity? and 2) what is the impact of requirement complexity regarding the development process? To address these questions, the research in this dissertation was conducted.

1.2 Dissertation Objectives and Contributions

In order to answer the question above, three main objectives are to be addressed in this dissertation to allow for an answer that can be scientifically supported and validated:

1. Evaluating and or understanding the structure and or dynamics of requirements
2. Quantifying and or measuring the complexity of requirements
3. Assessing the effect of the requirement complexity

The exact extent and precise contributions, as well as the formulation of hypotheses (Chapter 5), had to be defined in accordance with the state of the art and research, which required the comprehensive literature and publication reviews/studies that follow in Chapters 2 through 4. Nevertheless, answering the questions in Section 1.1 and addressing Objectives 1 through 3 means that this dissertation and its research content contribute a novel analysis approach that employs NLP to understand the structure of textual requirements in order to quantify and or assess complexity. The chain and logical structure for this contribution are shown in Figure 1.4:

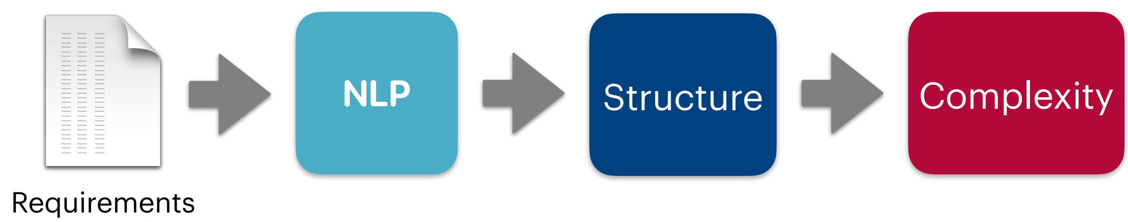

FIG. 1.4 - LOGICAL FLOW OF DISSERTATION CONTRIBUTION

Since the research of this dissection is potentially useful for practical system engineering and development processes, attention was paid to applicability, replicability, and scalability. These aspects were considered to ensure that the approach to develop and research to conduct were not only of scientific nature but could later also be used for applications and by practitioners. This relationship to practice is significant since the topic and problems mentioned above are of project management and financial nature, which have a critical position and significant influence in most development processes as well as missions overall.

1.3 Dissertation Content and Structure

While research and the exploration of something unknown, such as the development of a novel solution or approach, is inherently difficult to plan, this dissertation was structured following established methods (Blessing & Chakrabarti, 2009; Ehrlenspiel & Meerkamm, 2017) to guide the research process (also see Chapter 5 for more detail) and achieve the objectives defined in the previous section.

First, the state of the art and research are evaluated in a literature review. Since multiple topics converge here, the literature review addresses the fields of complexity **(Chapter 2)**, requirements engineering **(Chapter 3)**, and Natural Language Processing **(Chapter 4)** individually. The results and information provided by these chapters created the foundation for the definition of the research content as well as its distinction and uniqueness **(Chapter 5)**, which in turn includes the research hypotheses and anticipated results. In order to address these objectives, the actual development of novel solutions was conducted **(Chapters 6 & 7)**, which could then be applied and tested in case studies **(Chapters 8 through 10)** to provide data that allows for the validation of the hypotheses and research in general **(Chapter 11)**. The dissertation is concluded with a summary and outlook, which shows potential extensions and future research directions.

Figure 1.5 shows a visual overview with all the respective connections between the different chapters (excluding this introduction and the last chapter).

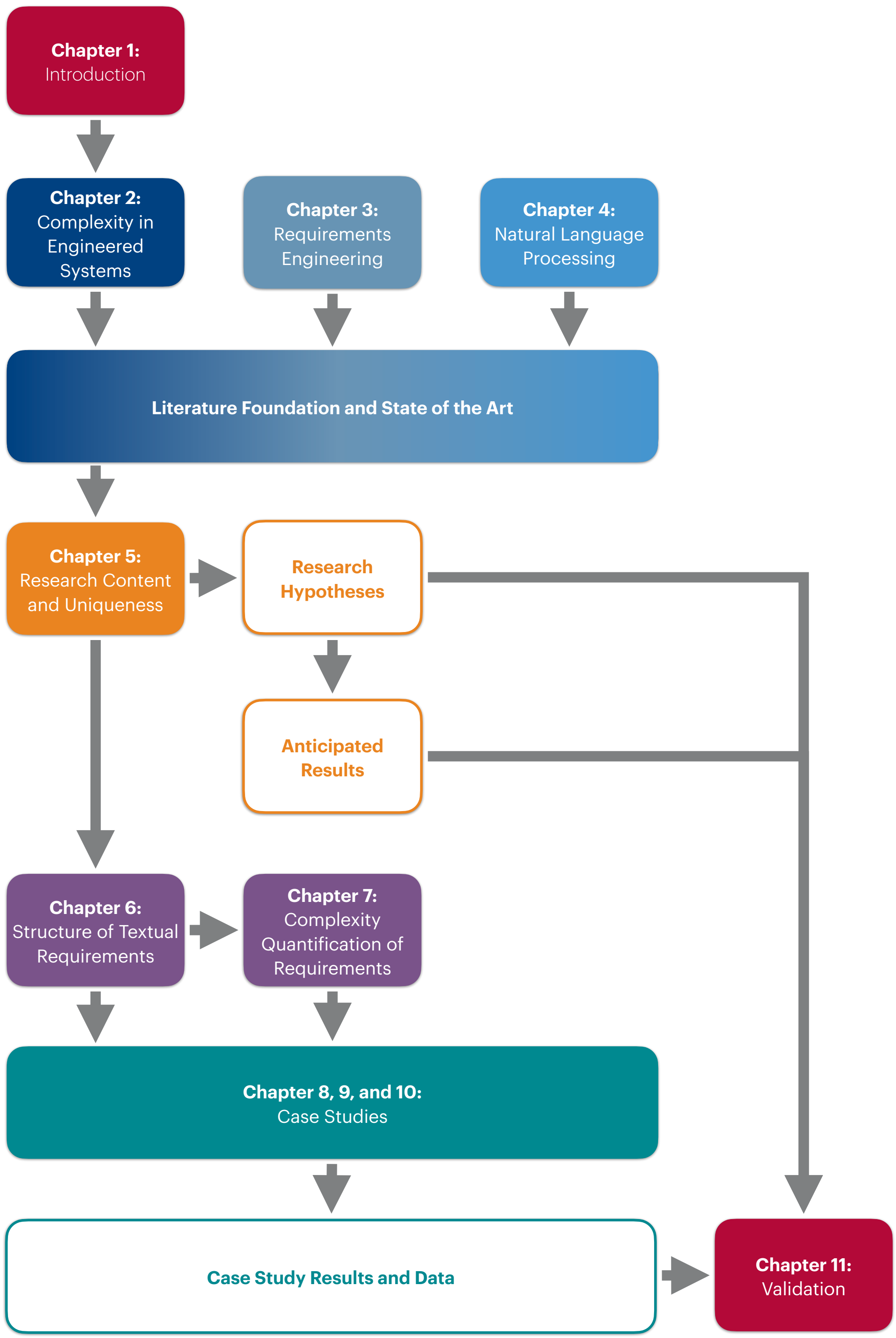

FIG. 1.5 - VISUAL SUMMARY OF DISSERTATION STRUCTURE

With the shown structured objectives, the research in this dissertation was conducted in the time between September 2019 and January 2022. All sources, material, and references are specified where indicated or in the appendices. Further information and data produced can be provided upon request.

The next chapter begins the evaluation of the state of the art and literature by covering the topic of complexity and its science.

BIBLIOGRAPHY OF CHAPTER 1

Blessing, L. T. M., & Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. London, United Kingdom: Springer-Verlag.

Boznak, R. G. (1994). When doing it right the first time is not enough. *Quality Progress, 27*(7), 74-78.

Boznak, R. G., & Decker, A. K. (1993). *Competitive Prod Dev: A Quality Approach to Succeeding in the 90s and beyond*. New York, NY: McGraw-Hill Education.

Chammard, T. B., Regalia, B., Karban, R., & Gomes, I. (2020). *Assisted authoring of model-based systems engineering documents.* Paper presented at the Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems.

DC-1 Request for Proposal. (2018). Retrieved from http://heroicrelics.org/prologue-rm/request-for-proposal/dscc8864.jpg.html

Ehrlenspiel, K., & Meerkamm, H. (2017). *Integrierte Produktentwicklung* (6th ed.): Carl Hanser Verlag.

Fruhlinger, J., Wailgum, T., & Sayer, P. (2020). 16 famous ERP disasters, dustups and disappointments. Retrieved from https://www.cio.com/article/2429865/enterprise-resource-planning-10-famous-erp-disasters-dustups-and-disappointments.html

Karban, R., Dekens, F., Herzig, S., Elaasar, M., & Jankevičius, N. (2016). *Creating system engineering products with executable models in a model-based engineering environment* (Vol. 9911): SPIE.

Kleedörfer, R. (1999). *Prozeß- und Änderungsmanagement der Integrierten Produktentwicklung*. Munich, Germany: Technical University of Munich.

Lindemann, U., Maurer, M., & Bran, T. (2009). *Structural Complexity Management*. Berlin Heidelberg, Germany: Springer-Verlag.

Lindemann, U., & Reichwald, R. (1998). *Integriertes Änderungsmanagement*. Berlin Heidelberg, Germany: Springer-Verlag.

Metropolitan Museum of Art. (2017). Carpenter's Adze from a Foundation Deposit for Hatshepsut's Temple. Retrieved from https://www.metmuseum.org/art/collection/search/547562

National Aeronautics and Space Administration (NASA). (2020). *NASA Systems Engineering Handbook*. Washington, DC.

National Aeronautics and Space Administration (NASA), & Jet Propulsion Laboratory. (2016). Europa Clipper spacecraft model. Retrieved from https://www.jpl.nasa.gov/spaceimages/images/largesize/PIA20025_hires.jpg

National Museum of American History. (2014). Alexander Graham Bell Experimental Telephone. Retrieved from https://americanhistory.si.edu/collections/search/object/nmah_689864

Sheard, S. A., & Mostashari, A. (2010). A Complexity Typology for Systems Engineering. *INCOSE International Symposium, 20*(1), 933-945. doi:10.1002/j.2334-5837.2010.tb01115.x

Wikimedia Commons. (2021). USAF Boeing B-52 Stratofortress. Retrieved from https://en.wikipedia.org/wiki/Boeing_B-52_Stratofortress

Wildemann, H. (2020). *Änderungsmanagement* (28th ed.). Munich, Germany: TCW.

CHAPTER 2: COMPLEXITY IN ENGINEERED SYSTEMS

"Weniger, aber besser."

(Less but better.)

Dieter Rahms

2.1 Content, Approach, and Sources

As mentioned in the introduction, this dissertation concerns and combines various scientific fields. The combination of these fields requires an assessment of the state of the art and research for each of them. Due to the extent and available literature for the topics, a separate chapter is dedicated to each of the three fields. Thus, Chapters 2 through 4 address the topics of complexity (Chapter 2), requirements (Chapter 3), and Natural Language Processing (Chapter 4). For each topic, important adjacent fields are also covered where applicable.

Overall, the literature reviews followed the method outlined by Budd, Thorp, and Donohew (1967), which describes the process of content analysis in accordance with the following steps: taxonomy development and keyword definition, keyword searches, review of relevant publications, evaluation of bibliography, synthesis and critiquing of the evaluated material.

The databases used for all literature searches were relevant to the domain of systems engineering and included private and public libraries, the Stevens Institute of Technology library, Google Scholar, INCOSE databases, IEEE Explore, Wiley's System Engineering repositories, the Design Society's repositories, Elsevier databases, as well as the Cornell University arXiv, TechRxiv, and engrXiv. All references and sources were documented, and the ones used in this dissertation are listed in the respective bibliographies at the end of each chapter unless otherwise specified. A wide range of sources was used, but peer-reviewed and higher quality publications were preferred to rely on sources with the highest credibility possible. Thus, the following hierarchy was applied for credibility: journals, books, conferences, scientific reports, pre-print/e-print/arXiv publications, presentations, and magazines. For reproducibility, all sources were cataloged as document files in a citation manager.

To provide the definitions and taxonomies that were used for the literature review, as well as for the dissertation as a whole, Table 2.1 shows the overview in accordance with the sections that were listed on the previous page.

TABLE 2.1 - TAXONOMY AND LIST OF KEYWORDS

Chapter	Element	Keywords
2	**Complexity/System Complexity**	complexity, complex systems, structural complexity, complexity classification, complexity categorization, complexity types, complexity interpretation
	System Complexity/Metrics	complexity metrics, complexity measures, complexity measurements, complexity quantifications
3	**Requirements Engineering**	requirements engineering, requirement management, requirements handling, change management, requirement evaluation, requirement assessment
	Requirement Structure/ Complexity	requirement structure, requirement architecture, requirement types, requirement complexity, requirement complexity metrics, requirement complexity measures, requirement complexity measurements, requirement complexity quantifications
4	**Natural Language Processing**	natural language processing, natural text processing, text processing, natural language understanding, natural language creation
	Natural Language Processing for Requirements Engineering	natural language requirements, textual requirements, requirement text processing, requirement interpretation, requirement text processing

2.2 General and Systems Complexity *

To begin the state of the art and section of the literature review on the topic of complexity, a general understanding of the term complexity shall be established. Due to the wide use of the term itself, a single search on Google Scholar for the term "complexity definition," for example, reveals a multitude of different interpretations and definitions, as also noted by Suh (2005). Furthermore, even in everyday conversations, the term complexity is used. Due to these different use cases, it is important to establish the baseline for interpretation.

The definition of complexity and framing of an interpretation can be a difficult task, though. Since not only has complexity science branched out and been adopted in a multitude of fields, as shown below, but the term complexity itself is often used with different and sometimes questionable interpretations. Thus, this section will take a brief look at the origins of complexity and then frame the term for the content of this dissertation.

Complexity was first mentioned by Weaver in 1948 (Weaver, 1948) and, over time, has led to the development of complexity science (Richardson, Cilliers, & Lissack, 2001). This science deals with the characteristics of complex systems that can be characterized by but are not limited to, emergent behavior due to reciprocities of system elements (Phelan, 2001), nonlinear and dynamic interactions of elements (Cilliers, 2000), and bilaterally dependent relations of elements (Strogatz, 2012); more about these characteristics below.

Weaver (1948) described two kinds of complexity: organized and disorganized complexity. The first category, organized complexity, is characterized by a substantial number of variables and "factors which are interrelated into an organic whole" (Weaver, 1948). These factors must be considered when the entire system is being analyzed. Problems of organized complexity differ from the ones pertaining to simplicity as they exceed small numbers of a few variables. The number of elements in these cases is still relevant, which distinguishes the organized from

---

* The content of this section has been extracted literally or with minor editorial modifications from M. Vierlboeck and R. Nilchiani (2022), "Requirement Engineering in the Age of System and Product Complexity – A Literature Review," published by IEEE.

the disorganized category. Disorganized complexity is characterized by an abundance of variables in a system. Each variable exhibits individual behavior, which is described as "erratic, or perhaps totally unknown" (Weaver, 1948). Despite all these individual influences, disorganized complexity tries to explain the behavior of the system in its totality and therefore allows for the analysis despite all the underlying variables. Such analyses are related to statistical techniques, which become applicable once individual behavior gives way to average behavior(s) that is assessed. Figure 2.1 below shows the two types and their characteristics.

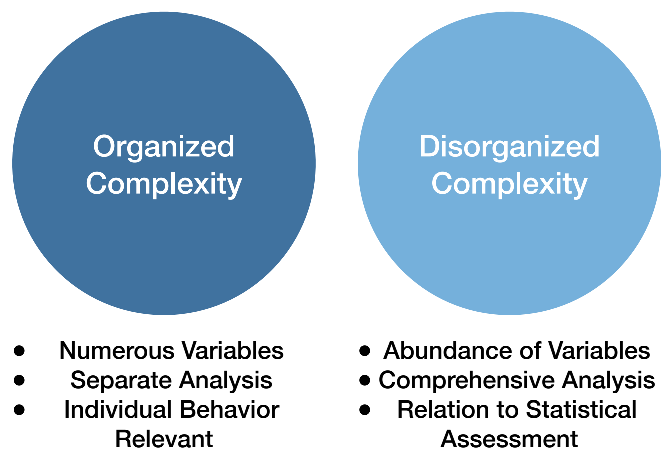

FIG. 2.1 - ORGANIZED AND DISORGANIZED COMPLEXITY AS PER WEAVER (1948)

With the listed attributes, general complexity is linked to the concept of emergence (S. A. Sheard & Mostashari, 2009), which occurs when an "increasing number of independent variables begin interacting in interdependent and unpredictable ways" (Sanders, 2003). This dependence on interactions and bilateral connections also prohibits the decomposition of the system into smaller, still functionally equivalent subsystems (DeRosa, Grisogono, Ryan, & Norman, 2008; Vandergriff, 2007).

In addition to the two types of complexity above, other terms have to be considered in the realm of complexity science to distinguish the term. First, complex has to be separated from complicated. The term complicated is what most people mean in everyday conversation when they describe something falsely as complex. In complexity science, a complicated system is characterized by possibly many components whose interactions and reactions are numerous, but their behavior, as well as the behavior of the system, is understood despite the sheer amount of aspects that might exceed the realm of a single human's understanding (Cotsaftis, 2009; Crawley, Cameron, & Selva, 2015; DeRosa et al., 2008; S. Sheard & Mostashari, 2011). Moreover, complex systems are characterized by concepts associated with the edge of chaos, multi-dependency dynamics, uncertainty, and emergence caused by the behavior and interaction of known components (Cotler, Hunter-Jones, Liu, & Yoshida, 2017; DeRosa et al., 2008; Phelan, 2001; Strogatz, 2003). Thus, while complicated systems can still be understood, complex systems lie beyond our understanding and, as a result, have to be handled differently.

Other scholars, such as Snowden & Boone (2007), have extended the distinctions above to include the terms chaos and simple, all centered around disorder. According to Snowden & Boone, their framework shall assist with the solution of a specific problem. The terms form areas in the framework and represent different dimensions, as shown in Figure 2.2.

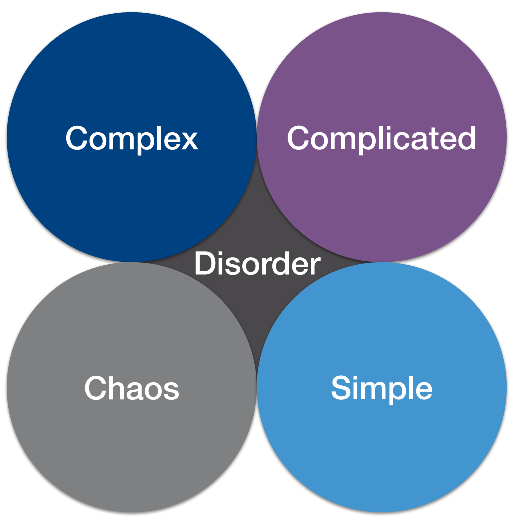

FIG. 2.2 - CYNEFIN FRAMEWORK WITH DIMENSIONS (SNOWDEN & BOONE, 2007)

The dimensions depicted are supposed to classify a given problem based on association. When a problem is simple, a shared understanding of the matter and problem exists, which allows for a comprehensive understanding of cause and effects. A complicated problem, as outlined above, can be understood through comprehensive analysis and evaluation since there is still a cause-and-effect relation, just not as obvious as in the simple realm. For complex problems, the relationship between cause and effect links is not visible anymore, which is why they are characterized by emergent properties, amongst other aspects (Cotsaftis, 2009). Sometimes, it is possible to evaluate the dynamics and interactions retroactively, but not always. In the chaotic space, patterns cannot be discerned at all, and no relationships can be identified. Should a chaotic state be reached, the goal cannot be the definition of a solution anymore but the return to one of the other domains.

With the variety and different terms and potentially fuzzy borders, it can be challenging to define complexity in a general way (Corning, 1998). Thus, for the content of this dissertation, the following description is used as a general interpretation:

Complexity is defined by the identification of the attributes & characteristics of complex systems. These attributes are the concepts of multi-dependency dynamics, uncertainty, and emergence, caused by the behavior and interaction of known components.

(Cotler et al., 2017; DeRosa et al., 2008; Phelan, 2001; Strogatz, 2003)

The interpretation above provides the frame for the complexity research of this dissertation. Based on this frame, the literature was assessed, especially regarding system complexity, design complexity, and metrics thereof, as this dissertation also includes the quantification of complexity. Thus, the following sections first address general complexity metrics (2.3) before looking into system design complexity in general (2.4) and lastly, specific metrics for system design complexity (2.5). Figure 2.3 below shows how the different chapters overlap and how they are related while also including the section numbers of this chapter for a better understanding and structural overview:

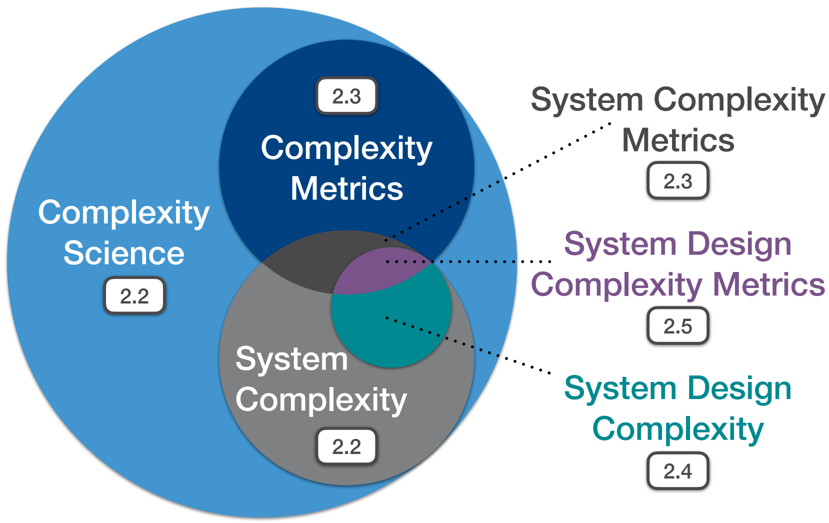

FIG. 2.3 - COMPLEXITY SCIENCE AND METRICS OVERVIEW WITH CHAPTER MARKERS

2.3 GENERAL COMPLEXITY METRICS *

In order to characterize and structure the research, the evaluated metrics were classified and assigned to different engineering/science fields. This allows for a simplified differentiation, and furthermore can foster the transfer of other concepts from and to those fields. The applied and assigned fields/categories were derived from general domains of engineering in relation to systems (Mar, 1997; "The Merriam-Webster Dictionary, International Edition," 2016). Based on the application of systems engineering in various fields, publications describe the main branches to consider as civil, mechanical, electrical, and chemical engineering (Hamilton, 2000). These are in accordance with the first edition of the Oxford Handbook of Interdisciplinarity (Frodeman, Klein, Mitcham, & Holbrook, 2010), which adds industrial engineers as a fifth branch. In the

* The content of this section has been extracted literally or with minor editorial modifications from M. Vierlboeck and R. Nilchiani (2022), "Requirement Engineering in the Age of System and Product Complexity – A Literature Review," published by IEEE. Copyright transferred to IEEE. © 2021 IEEE

most recent edition, even more diversity can be found, and the following branches are listed as engineering fields: civil, mechanical, chemical, electrical, electronic, industrial, nuclear, computer, biological, and nano (Frodeman, Klein, & Dos Santos Pacheco, 2017). This is due to subdivision over the years, as described by Dandy et al. (Dandy, Walker, Daniell, & Warner, 2008). The list can be regarded as sufficient for the purpose of this dissertation from the engineering side. However, since the research is based on engineered systems, the systems side also adds to the possible domains that are to be considered.

Solely from a wording perspective, the term system itself brings up a possible area: systems engineering. While containing the term engineering, it does not necessarily always show the above-mentioned definitions (Mar, 1997; "The Merriam-Webster Dictionary, International Edition," 2016) and is considered a "transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems" (INCOSE, 2019). Thus, systems engineering directly works with the systems in question of the research at hand and has to be therefore included. The so-called "transdisciplinary" (INCOSE, 2019) aspect of systems engineering then allows for the deduction of fields that are affected by it and therefore cohere with engineered systems.

Looking at the branches and fields connected to systems engineering, besides the ones already covered above, we see that systems engineering is linked to many domains not necessarily strictly considered engineering. This diversity is also the reason for sources and references mentioning different extents (Kossiakoff, Sweet, Seymour, & Biemer, 2011; "Systems Engineering Across Multiple Domains," 2014) and Kossiakoff et al. even describe the domains as expanding (Kossiakoff et al., 2011). Systems engineering can generally be applied to many areas under certain conditions as long as the area "develops critical and logical thinking" (INCOSE, 2023). General categories have been defined and include the domains of management, engineering, technical, political & legal, human, and social (Kossiakoff et al., 2011). Due to the vast amount of possible areas and domains, a generic approach has been chosen to allow for structured research. Instead of having to consider each possible domain, the generic categories listed above were assessed to also cover all subdomains.

Still, due to the abundance, differences, and amount of complexity measures and research across the fields and areas mentioned in the paragraphs above, it would be impractical to attempt listing every single relevant publication and work in existence. For example, the conjunct search terms "chemical engineering" and "complexity" yielded over 1.5 million results on Google Scholar in February 2020, which has since then grown to 2.1 million in October of 2022; a similarly high number was found for "mechanical engineering" combined with "complexity," and "industrial engineering" in conjunction with "complexity" yielded over 3.4 million results (October 2022). Thus, a guided and structured approach for the actual summary and elicitation of the information is critical.

The chosen approach used citation counts as a measure to rate and assess publications. Based on this approach, the most cited publications regarding the term and measures of complexity were researched in a first step. In a second step then, the publications citing these works were searched for further measures and or metrics based on these terms in conjunction. In order to define which publications are popular enough to be considered, a hard cut-off was chosen at a citation count of 250. This reduced the resulting lists to a reasonable amount and also allowed the extraction of popular approaches. The process was then repeated iteratively for each discovered publication in order to create a network of citations based on popular results that fit the topic and work at hand. Due to the fact that citation counts can rise over time, the possibility remained that recent publications exist, which might be eliminated by the 250 count due to their lack of age despite being important and or applicable. Thus, a subsequent round of literature research was conducted only for the time between 2010 and 2022 with all the parameters above but a citation count limit of 100.

The above-described approach for the literature research was repeated for the perspective of the engineering fields as well as systems engineering on its own. The results are described hereinafter. Furthermore, the connections between the different publications were evaluated by linking them to each other via cross-references, which created a publication map depicting the overall realm and influences.

Based on the described analyses, a final evaluation was conducted to derive a comprehensive structure. This includes a general overall map based on all the mentioned aspects and approaches as well as their connections and interactions. The classifications derived and defined connect the overall structure and show the interfaces, whereas the degree of complication and focus therein define the perspective. The map could then be used to assess the different fields, which were deduced and defined above, leaving a set of publications to differentiate the fields. The most important publications are outlined before presenting the resulting map and overview.

When looking for complexity metrics, one of the first and often cited publications that addresses complexity in a scientific way emerges: Shannon's "Mathematical Theory of Communication," which is a fundamental work of the complexity research regarding entropy (Shannon, 1948). Shannon explains that the entropy of a system describes the set of probabilities said system has regarding its state. Therefore, a metric based on the described entropy is possible. Albeit not necessarily directly related to any field, Shannon's research is based on communication and signal processing.

Another popular publication, possibly the most popular publication regarding complexity measures, is McCabe's 1976 paper which describes a graph-theoretic complexity measure (McCabe, 1976). In said publication, the author outlines the connection of graph-theory concepts and complexity and connects them to the design/structure of computer programs as well as their development. Based on this association, the research belongs to the computer science field.

The third and last popular publication, linked to over 9,400 (as of October 2022) other publications according to Google Scholar, is Kauffman's 1996 book "At Home in the Universe" (Kauffman, 1996). In this book, Kauffman describes and explains self-organized complexity and relates it to various biological structures, such as living organisms.

Based on these three starting points, other publications were connected to them with the two citation count limits mentioned on the previous page. Every publication was assigned to one of the aforementioned fields. This yielded a map and overview, showing the pervasion of the different complexity measures in the different fields: Signal Complexity (Electrical Engineering), Physical Complexity (Chemical, Nuclear, and Nano Engineering), Infrastructure and Network Complexity (Civil, Industrial, and Electrical Engineering), Biochemical Complexity, Design & Manufacturing Complexity (Industrial, Mechanical, and in part Electrical Engineering), Software and Code Complexity (Computer Engineering). All of these areas show significant research and measures/metrics regarding complexity. Furthermore, the defined engineering categories outlined in the previous chapter can directly be sorted to fit into each of the categories, even though there is overlap for some.

The full list of discovered research publications can be found in the bibliography of Chapter 2, with the citations listed in Table 2.2 and Table 2.3. Herein, over 30 publications were found that either contain, address, or otherwise pertain to complexity metrics and or measures.

All of these references were mapped out as described and thus yielded the fields listed in the previous paragraph. With the overview of the references, the designed map resulting from their connections yielded the map depicted in Figure 2.4. Some publications possibly touched more than one field, but in general, these areas were discernible with the adjacencies shown.

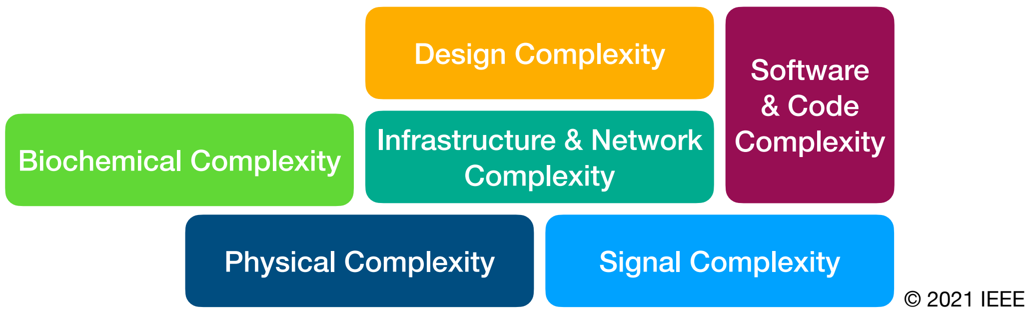

FIG. 2.4 - COMPLEXITY AREAS AND SCIENTIFIC FIELDS (VIERLBOECK & NILCHIANI, 2021)

Following the general complexity measures, a specific look at system design and development complexity was taken in order to provide a comprehensive and detailed overview for the field and topics as well, which also included the metrics as shown in Figure 2.4.

Table 2.2 - Complexity Areas and Scientific Fields References - Biochemical and Design Complexity

Author(s)	Title	Area
S. A. Kauffman	At Home in the Universe	Biochemical Complexity
C. Adami, C. Ofria, T. C. Collier	Evolution of biological complexity	Biochemical Complexity
D. W. McShea	The hierarchical structure of organisms	Biochemical Complexity
P. Romero, Z. Obradovic, X. Li, E. C. Garner, C. J. Brown, A. K. Dunker	Sequence complexity of disordered protein	Biochemical Complexity
D. Bonchev, D. H. Rouvray	Complexity in Chemistry, Biology, and Ecology	Biochemical Complexity
R. M. Hazen, P. L. Griffin, J. M. Carothers, J. W. Szostak	Functional information and the emergence of biocomplexity	Biochemical Complexity
C. P. Panos, K. C. Chatzisavvas, C. C. Moustakidis, N. Nikolaidis, S. E. Massen, K. D. Sen	Atomic Statistical Complexity	Biochemical Complexity
P. R. Bryant	The order of complexity of electrical networks	Design Complexity
H. A. Bashir, V. Thomson	Estimating Design Complexity	Design Complexity
C. Eun Sook, K. Min Sun, K. Soo Dong	Component metrics to measure component quality	Design Complexity
R. Subramanyam, M. S. Krishnan	Empirical analysis of CK metrics for object-oriented design complexity	Design Complexity
H. A. Bashir, V. Thomson	Estimating design effort	Design Complexity
C. C. Bozarth, D. P. Warsing, B., B. Flynn, E. J. Flynn	The impact of supply chain complexity on manufacturing plant performance	Design Complexity
F. Isik	An entropy-based approach for measuring complexity in supply chains	Design Complexity
J. D. Summers, J. J. Shah	Mechanical Engineering Design Complexity Metrics	Design Complexity
W. ElMaraghy, H. ElMaraghy, T. Tomiyama, L. Monostori	Complexity in engineering design and manufacturing	Design Complexity

TABLE 2.3 - COMPLEXITY AREAS AND SCIENTIFIC FIELDS REFERENCES - INFRASTRUCTURE THROUGH SOFTWARE

Author(s)	Title	Area
J. Portugali, H. Meyer, E. Stolk, E. Tan	Complexity Theories of Cities Have Come of Age	Infrastructure & Network Complexity
S. H. Strogatz	Exploring complex networks	Infrastructure & Network Complexity
S. Lloyd, H. Pagels	Complexity as thermodynamic depth	Physical Complexity
M. Gell-Mann, S. Lloyd	Information measures, effective complexity, and total information	Physical Complexity
C. E. Shannon	A mathematical theory of communication	Signal & Information Complexity
H. A. Simon	The Architecture of Complexity	Signal & Information Complexity
J. H. Holland	Hidden Order	Signal & Information Complexity
J. M. Carlson, J. Doyle	Complexity and robustness	Signal & Information Complexity
T. J. McCabe	A Complexity Measure	Software & Code Complexity
E. J. Weyuker	Evaluating software complexity measures	Software & Code Complexity
S. R. Chidamber, C. F. Kemerer	A metrics suite for object oriented design	Software & Code Complexity
V. R. Basili, L. C. Briand, W. L. Melo	A validation of object-oriented design metrics as quality indicators	Software & Code Complexity
C. Collberg, C. Thomborson, D. Low	A Taxonomy of Obfuscating Transformations	Software & Code Complexity
A. E. Hassan	Predicting faults using the complexity of code changes	Software & Code Complexity
J. P. Crutchfield, K. Wiesner	Simplicity and Complexity	Software & Code Complexity
N. Fenton, J. Bieman	Software Metrics: A Rigorous and Practical Approach	Software & Code Complexity

2.4 System Design & Development Complexity - State of the Art *

When researching the topic of system design complexity or complexity within system design, Google Scholar yields 5.5 million results (as of October 2022). This high number is, at least in part, caused by the fact that system complexity is subject to a high degree of interpretation, just like general complexity (see Section 2.2). Whereas standards exist in requirements engineering, for example, system complexity is interpreted differently in various fields and by individual authors/researchers (Jacobs, 2007). For instance, Baldwin & Clark describe complexity in system design as proportional to design decisions and parameters (Baldwin & Clark, 2000). Griffin describes design complexity as the number of functions in or performed by a product (Griffin, 1997a, 1997b), and other authors define system design complexity as the number of parts and or individual parts in a product/system (Brooks, 1987; Gupta & Krishnan, 1999). Moreover, within other fields, such as project management or supply chain research, for example, additional variations and interpretations can be found (Jacobs, 2007; Jacobs & Swink, 2011).

With this abundance of different interpretations and definitions, plus the above-shown inconsistencies and at least partial discrepancies, the literature research regarding system design complexity becomes complicated as inclusion of all the different adaptations would not be feasible nor expedient. In addition, utilizing a set definition for system design complexity based on the literature review would already limit the characteristics of the research scope to the chosen interpretation, which, given the nature of the proposed work, potentially interferers with the creation of a novel problem-solving approach and framework. Thus, it was decided not to chose a set definition to retain all options while the literature review and foundation could still be evaluated. Yet, by limiting the scope for the term complexity to the above-mentioned system

* The content of this section has been extracted literally or with minor editorial modifications from M. Vierlboeck and R. Nilchiani (2022), "Requirement Engineering in the Age of System and Product Complexity – A Literature Review," published by IEEE.

design/development, the field was manageable but not restricted enough so that important publications would be missed.

With the general aspects of complexity described in Section 2.2, a look at the literature regarding complexity in system and product development/design shows that in these fields, the term complexity was diversified according to specific parts of the development process of the system. Göpfert (1998) described that system development complexity showed two facets: technical complexity and organizational complexity. Göpfert stresses that these two types of complexity cannot be seen as separate as they influence each other and have to be considered together in order to handle them appropriately (Göpfert, 1998). This bisectional partition is also found in other, not directly system development related fields (Heylighen, 1999), which the next approach also relates to.

Weber (2005) proposed an approach similar to the previous one by Göpfert. Weber's complexity interpretation for the field of product/system development and engineering design included five dimensions that are directly related to strategic components, as shown in Figure 2.5 (Weber, 2005): numerical complexity, relational/structural complexity, variational complexity, disciplinary complexity, and organizational complexity.

In addition to defining the dimensions, Weber also divided them into two overarching groups: the product/system category, which encompasses the first three dimensions, and the process category, which includes the last two. This overall forms the constellation shown in Figure 2.5 on the next page.

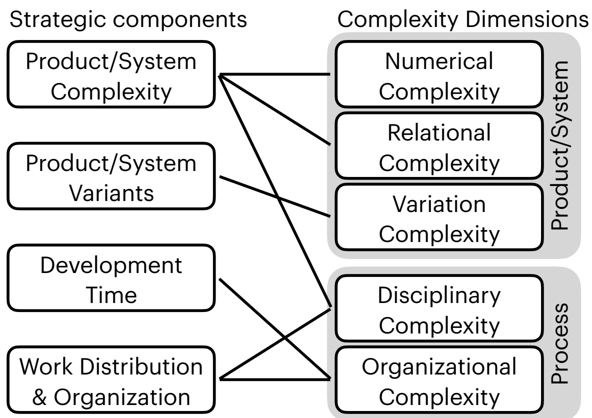

Fig. 2.5 - Complexity Dimensions and Corresponding Strategic Components by Weber (Weber, 2005)

As a reaction to the approach by Göpfert, Weber also distinguished between the productrelated complexity dimensions and the process/organizational aspects (Baccarini, 1996). Although Weber does not directly approach the technical complexity defined by Göpfert (Göpfert, 1998), their approach poses an extension of the former work and builds upon it.

In 1998, Braha & Maimon also addressed design complexity from a mathematical perspective and described two categories of complexity for the Formal Design Theory (FDT) (Braha & Maimon, 2013): structural complexity and functional complexity. The former describes the complexity related to the representation of information, meaning artifacts/elements and what they represent; the latter outlines complexity regarding the notion behind information, regardless of how it is represented. This implies that a function can be independent of the elements it is comprised of, whereas a structural aspect relates to the actual elements and components. This definition of structural complexity is similar to the next publication.

In 2009, Lindemann et al. (Lindemann, Maurer, & Bran, 2009) proposed a different approach which they also name "structural complexity," but without any relationship to the terms previously described. It is important to note that the term "structural complexity" used by Lindemann et al. is not solely related to spatial structure as described in other fields, for example

(Heylighen, 1999). Instead, the term relates to all "dependencies within the elements in technical systems" which form structures and cause complexity (Lindemann et al., 2009). As for influencing factors, Lindemann et al. described four major fields that shape and form the structural aspects and therefore impact complexity: market complexity, product complexity, organizational complexity, and process complexity.

All the above-listed interpretations show that there are various ways to approach the complexity in system development and design. This becomes even more apparent in the next section, which looks at metrics and characteristics of these manifestations in particular. Regardless, all four of the above-mentioned publications outline three key aspects:

I. The complexity of the system is connected to, but separate from the complexity of the development process and organizational aspects.

II. Complexity can exist within the functions of a system but also within the architecture and structure.

III. Various other factors can influence the design and development complexity, and these aspects can even possess complexity of their own.

With the above-described conclusions, the researched literature basis can be conflated to allow for an interpretation that connects all of them. Thus, the following dimensions for complexity form the basis for the research in this dissertation:

I. **Functional Complexity** represents the complexity behind the functions of a system or product. Functions are not necessarily related to an actual element or component but solely describe the purpose, notion, and what the system does. This type of complexity is similar to what some authors describe as dynamic complexity of a system (Sinha, 2014; Sinha & Suh, 2018).

II. **Structural Complexity** represents the complexity resulting from the dependencies, connections, and interactions between components within the system.

III. **Organizational and Process Complexity** represents the complexity originating from the process and organizational architecture around and behind the system. Thus, this last dimension of complexity is directly related to the actual system development and approach thereof.

These three dimensions are visualized in Figure 2.6, showing their coexistence and relations. Herein, the organization and process is depicted as the overarching construct for the actual product structure and functions. The latter two directly interact as the functions of the product imply the elements and components, which then again facilitate and enable the functions. This results in the two halves of the lower space, whereas the organizational aspects accompany and overarch both. In addition, these spaces can also be dynamic as they are not necessarily unchangeable over time, as indicated by the arrow shape of the organizational and process component. Despite the dynamic thought, the dimensions as groups are constant and always applicable.

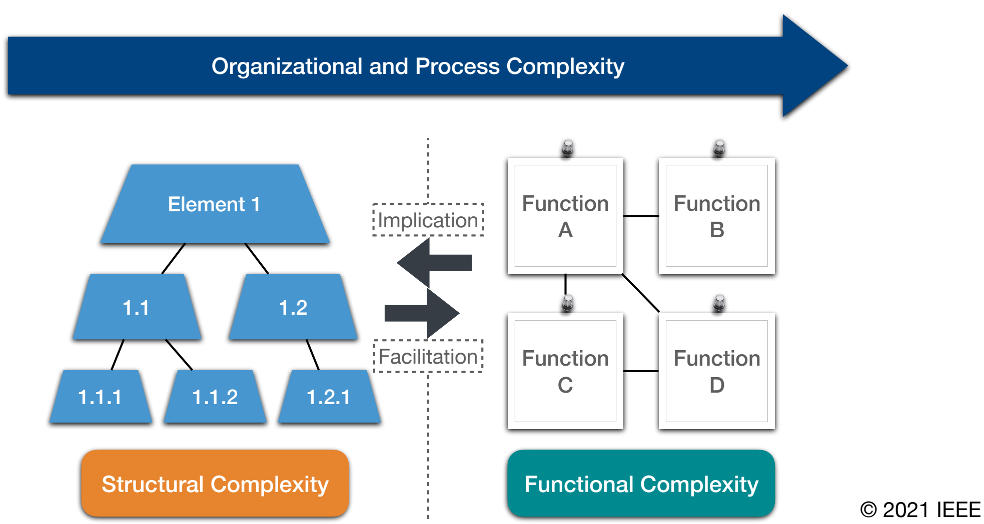

FIG. 2.6 - DIMENSIONS OF PRODUCT DESIGN AND DEVELOPMENT COMPLEXITY (VIERLBOECK & NILCHIANI, 2021)

Now, looking at the most recent publications and trends in the field of system design and development complexity, a few major trends and publications can be seen.

First, the most recent work of Sinha (Sinha, Han, & Suh, 2020; Sinha & Suh, 2018; Sinha, Suh, & de Weck, 2018) shows research regarding the complexity analyses of modular systems and shows the application of various neoteric approaches, such as Pareto-Optimization (Sinha & Suh, 2018) and System Clustering Algorithms (Sinha et al., 2020) to address the issue of structural complexity and system architecture. With the proposed framework Sinha & Suh claim that "complex systems can be optimized for degree of modularity, while variation of structural complexity among system modules is minimized" (Sinha & Suh, 2018). Figure 2.7 below shows a case study presented by said authors for bogie modularization configurations. The plot depicts the different modularity parameters and provides possible results in comparison to a reference design indicated by the red dot in the middle:

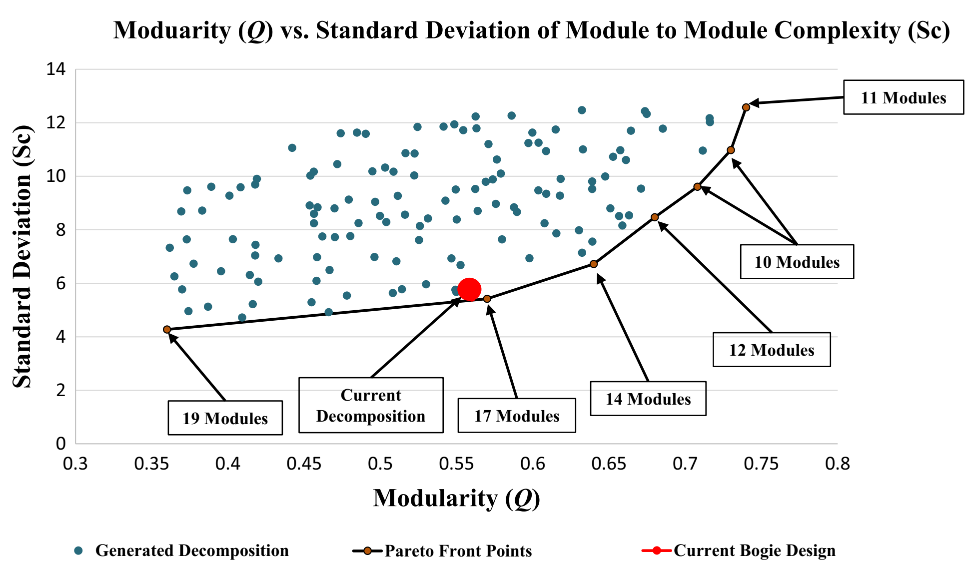

FIG. 2.7 - DIMENSIONS OF SYSTEM DESIGN & DEVELOPMENT COMPLEXITY (SINHA & SUH, 2018)

Sinha & Suh also stress the lack of quantifiable approaches for system complexity in the design phase (Sinha & Suh, 2018). Overall though, these publications, while along the lines of the research in the dissertation, target specifically the design phase, which is situated after the requirement definition and even the function definition (more about these circumstances is described in detail in Chapter 5 of this dissertation).

Second, another trend seen is the application of various novel algorithms and tools to model complexity dimensions, such as structural or organizational complexity. Examples of these approaches/tools include agent-based modeling (Benabdellah, Bouhaddou, & Benghabrit, 2019), statistical methods (Vogel & Lasch, 2018; Vyron, Panos, & James, 2018), and mechanism-based equifinal causal relations evaluation (Sihvonen & Pajunen, 2019).

Third, multiple recent publications were discovered that tackle the topics of new system development and the influences therein regarding complexity in all three aforementioned dimensions. These publications (Açıkgöz, Günsel, Kuzey, & Seçgin, 2016; Benabdellah et al., 2019; Cui & Wu, 2017; Fain, Žavbi, & Vukašinović, 2016; Mauerhoefer, Strese, & Brettel, 2017; Sihvonen & Pajunen, 2019) apply new approaches and research results to the complexity of system development in order to deduce, predict, and or manage the success of new products/ services to be developed. As a result, these resift contributions address the whole development process, but in all assessed cases, the approach begins with the design phase when requirements are transformed into design specifications (Benabdellah et al., 2019). Therefore, while these approaches are novel and utilize new tools, they do not address the area of research in this dissertation, as outlined in Chapter 5.

Fourth, the last trend identified was a trajectory less related to complexity itself but nevertheless connected to it: sustainability. Multiple publications by different authors (Katsikeas, Leonidou, & Zeriti, 2016; S. Kim & Moon, 2019; Schöggl, Baumgartner, & Hofer, 2017) were found that addressed the topic of sustainability and its implications also with regard to complexity. While these research publications do not necessarily contribute to the topic of system and development complexity, they do add a new facet and influencing factor that has to be considered down the line, as the implications of sustainability can cause major disruptions within all three of the system complexity dimensions (see Figure 2.6).

In addition to the four general trends described above, two publications were identified that are closely related to the work in this dissertation. Said two papers by Yang et al. (Yang, Shan, Jiang, Yang, & Yao, 2018) and Malmiry et al. (Malmiry, Pailhès, Qureshi, Antoine, & Dantan, 2016) both show ideas in line with the ones described for the research in this dissertation, albeit they target different aspects of the system development process. These two publications will be analyzed in detail below.

The first mentioned publication was released by Yang et al. in 2018 (Yang et al., 2018). The paper brought together the topics of project/organizational complexity in system development, customer needs, and entropy. The authors describe their approach to assess and manage the organizational complexity of new system development projects based on information entropy and the preferences of customers. Yang et al. apply the PageRank algorithm to define the importance of needs and additionally fed the results into a Design Structure Matrix approach to address the management of project organization complexity. Thus, the work of Yang et al. bears similarities to the one shown in this dissertation but is limited to customer needs as well as organizational complexity. The functional side of the system development was tackled by Yang et al. solely to enable the evaluation of importance of customer needs.

The second mentioned publication was released by Malmiry et al. in 2016 (Malmiry, Pailhès, et al., 2016). It addresses the issue of epistemic uncertainty (Turner & Cochrane, 1993) within the design process. The authors address this uncertainty by providing a structural decomposition of the product and its functions. Mallory et al. combine this decomposition with energy flow and interconnections between the required functions (Malmiry, Dantan, Pailhès, & Antoine, 2016) to allow the derivation of a structural assessment. With these insights, Malmiry et al. provide a quantifiable evaluation of a system function and structure which, through the decomposition, permeates all necessary functions as well as structural levels.

With the quantifiable evaluation, the approach by Malmiry et al. supports decision-making on each level and therefore can reduce complexity. It has to be noted though, that this approach requires an already existing or developing functional structure. The authors thus provided a viable approach to reduce the epistemic uncertainty throughout the progress of the development but did not connect nor characterize multiple interacting requirements in the requirements definition phase. The utilized tools are potentially applicable to the research in this dissertation and were be considered during the execution.

Figure 2.8 shows an exemplary structure analysis by Malmiry et al. (Malmiry, Pailhès, et al., 2016). The level of Characteristics-Properties Modelling (CPM) is added to the structure and functional levels to allow for an analysis of energy flow in the product. As depicted, the CPM level incorporates and represents a combination of different function and structure elements to allow for an assessment of the energy movements in the system. With this representation and underlying methods, the connections can be assessed and even sensitivity analyses performed. This enables insights into the connections and how the different functions cooperate and interfere with the structural elements.

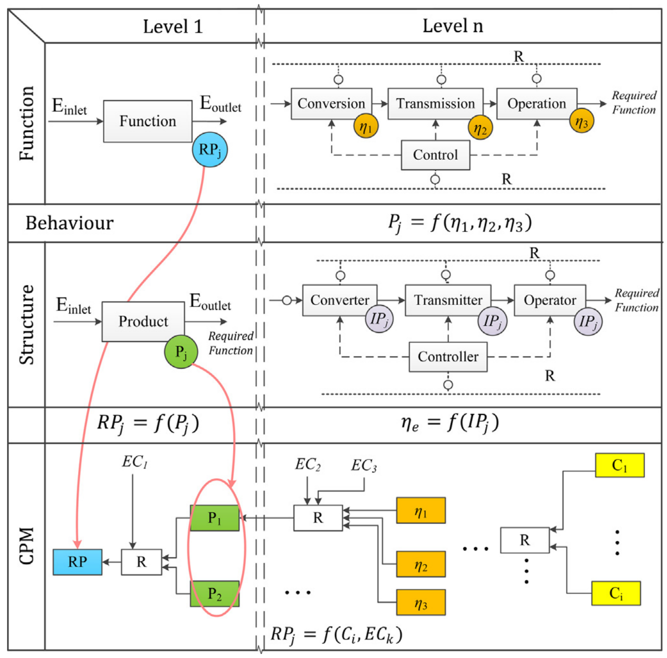

FIG. 2.8 - MODELING APPROACH WITH CPM BY MALMIRY ET AL. (2016)

Both of the above described research results show the potential and that an address of complexity even before the defining design phase is possible with promising merits. This supported the auspicious character of the work in this dissertation and also the later proposed research hypotheses (see Section 5.1).

2.5 SYSTEM DESIGN & DEVELOPMENT COMPLEXITY METRICS

As discussed, the complexity space is vast and the term itself can be interpreted in many different ways. This leads to the same problem for this section since the same predicament applies to the system design complexity metrics: while there are various metrics for complexity, they are not all necessarily applicable in their entirety to the topic of system development and design. For example, the complexity measure by McCabe (1976) is strongly related to software and control programs, as described in the previous section, and while this measure is applicable to these types of systems, it cannot be easily transferred to another system that does not necessarily inherit information paths for example, as this characteristic is a crucial component to McCabe's approach (Albrecht, 1979; Halstead, 1977; Henry & Kafura, 1981; Oviedo, 1980). Therefore, for this literature research, the results taken into consideration were the ones specifically targeting products and or systems in their entirety combined with their design/ development. In accordance with the areas described by ElMaraghy et al. (2012), this limits the complexity of the system to develop. Thus, the manufacturing complexity as well as marketing and other business-related complexity aspects are excluded. This frame limits the literature to a manageable amount and, in addition, provides a good initial connection to the topic of this dissertation work while not restricting the contribution.

The first type of relevant complexity metrics originates from Griffin and Kannapan (Griffin, 1993; Kannapan, 1995). Griffin (Griffin, 1993) initially developed a metric for development cycle time performance and noticed a correlation with system complexity based on functions. Based on these results, the author deduces that the developed metric could also be transferred and used to determine system complexity. The second mentioned publication, by Kannapan (1995), bases its complexity measure on component and system relationships regarding their intent and therefore derives metrics for categorization.

Unfortunately, while applicable, the focus on functions by the above-mentioned publications is very limiting since there are more dimensions to system complexity, as described in the previous section. These limitations are also outlined by Dierneder & Scheidl (2001). Thus, building upon Griffin (1993) and Kannapan (1995), Bashir & Thomson (1999) extend the metric based on functions by decomposition and therefore introduced a structural aspect. With this combination, it is possible to deduct the complexity and design effort based on the function tree, for example.

Dierneder & Scheidl (2001) further build upon the mentioned efforts as they described several aspects that they deemed to be inconsistent, such as location influence of functions in the tree of Bashir & Thomson (1999). Thus, Dierneder & Sheidl (2001) developed a complexity measure based on three metrics: Functional System Complexity, Technical System Complexity, and Reliability Complexity. All three metrics are defined with equations that yield concrete and quantifiable results and allow for a precise assessment of the complexity but require thorough functional decomposition and detailed specification.

Another measure for system design complexity, which was also referenced by authors mentioned in the previous paragraph, was provided in 1999 by Suh (Nam P. Suh, 1999). In this publication, the author describes an approach to quantify complexity in a time-independent and time-dependent way. For the former, Suh splits the complexity of a system into real and imaginary complexity and calculates the sum as the absolute complexity. To obtain quantifiable values, Suh looked at the function aspects and juxtaposed them with the design aspects to define which functions required more design parameters to be fulfilled. For the time-dependent complexity, the author distinguishes combinatorial and periodic complexity. For the former, a system will increase its complexity over time with accumulated information, whereas for the latter, the system can reset and renew in periodical phases. Suh also provides case studies for the developed approach and shows that the measures are applicable in a static and dynamic way.

The next relevant publication is the dissertation of Craig Read (2008). In their dissertation, the author describes a framework for complexity characteristics and measurements (see Figure 2.8 below). They distinguish between Problem Types, Coping Mechanism Types, Cause Types, Concepts and Classification Types, Definition Types, and Measurement Types - all in conjunction with complexity. These aspects influence each other bilaterally, according to Read. The connections are depicted in Figure 2.9 on the next page.

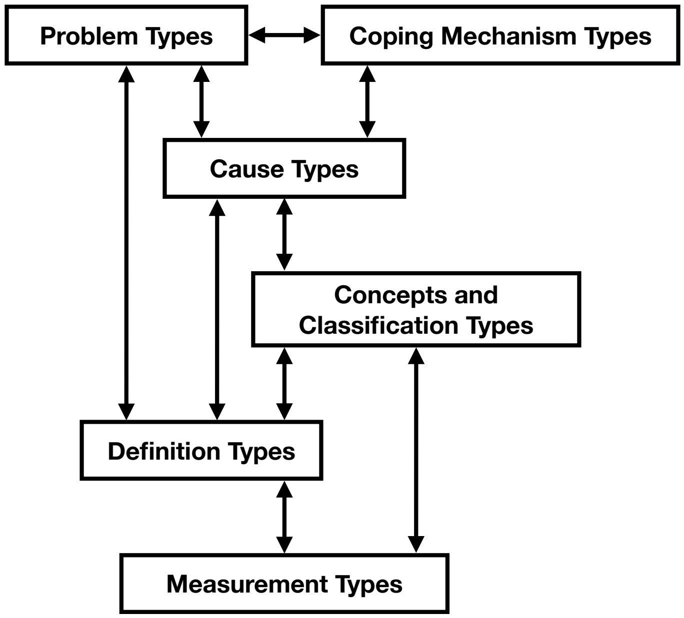

Fig. 2.9 - Complexity Characteristics Relationships by Read (2008)

All these characteristics of complexity are bound together by Read into a framework that allows for a structured and objective assessment (Read, 2008). Read also applied and tested the metric in various case studies and deemed the creation of the metrics successful. Yet, Read stresses that the future integration of the measurement approach and further expansion of the classification framework is critical. Overall, Read proved the success of a holistic complexity measure for engineered systems, which supports and provides a good foundation for the research and contribution in this paper.

Chronologically advancing, Sinha & de Weck presented an approach and quantification for structural complexity in 2013 (Sinha & de Weck, 2013). In this approach, the authors base their measure of total complexity in a product on the individual parts in conjunction with the links and interfaces between the parts. Therefore, the measure combines a combination of architectural as well as non-architectural aspects. The authors then combine the developed structure with topological complexity measured via matrix entropy to achieve their measure.

Sinha & de Weck also conclude that system and or product structure complexity correlate strongly with information complexity measures and metrics for distributed networks.

Another relevant concept, which also shows its importance again below in the most recent trends, is the association of cost/economics and system complexity. The connection between these two fields and topics has led to the creation of specific and general complexity characteristics as well, as described by Orfi, Terpenny, & Sahin-Sariisik (2011, 2012). In said publications, the authors describe general complexity factors that originate from the design and development and impact/influence the economics of the system. Five derived dimensions are defined: the structural index, design index, functional index, production index, and variety. These factors are linked by the authors to various cost drivers in order to assess the implications of each dimension. Overall, this approach proposes another general system complexity measure and metric, but it targets a slightly different area/level compared to the research in this dissertation, which targets the development of a system and does not include the overarching structure or product families that the metric by Orfi, Terpenny, & Sahin-Sariisik (2011, 2012) relies on. The implications of a product family and or variety are considered in this dissertation, but the main focus of the research does not primarily target these areas of the system development. This focus stems from the fact that requirements are usually defined for one development process in which variety can induce restrictions at best.

Looking at the most recent contributions published regarding the topic and field of system design complexity metrics and or measures, no approach holistic and comprehensive like the ones described above were found. The foci of the work and publications recently released (in the last five years, as of August 2022) all targeted specific topics and fields but did not provide comprehensive and or field overarching approaches. Nevertheless, the discovered trends and foci are summed up and described hereinafter.

First, a focus on system overarching attributes was discovered, which looks at the complexity and causations behind multi-variant structures and the implications thereof (Ghosh, Kristjandottir, Hvam, & Vareilles, 2018; G. Kim, Kwon, Suh, & Ahn, 2016; Mourtzis, Fotia, & Boli, 2017; Park & Okudan Kremer, 2019; Xiao, Zhou, & Sheng, 2016). For example, Kim et al. (2016) assessed the structural and architectural complexity of families and platforms, and Park & Okudan Kremer (2019) looked at the network topology of the structures of such families. While some of the mentioned and discovered publications provide metrics and or characteristics, they are closely and solely related to the variability or, in some cases, adaptability of the structure and therefore not universally or at least not widely applicable. Therefore, while valid, these works are not relevant for the contribution in this dissertation.

Second, a similar trend to the topics above was found to be the application of novel and neoteric computer aided approaches attempting to address system complexity and or the handling thereof (Gonzalez Castro, Panarotto, Borgue, & Isaksson, 2020; Raja, Kokkolaras, & Isaksson, 2019; Schuh, Rudolf, & Mattern, 2016; von Bary, Rebentisch, & Becerril, 2018). For example, von Bary, Rebentisch, & Becerril (2018) simulated the implications of organizational complexity on the value of a system development project based on an agent-based simulation. Schuh, Dölle, & Koch (2018) proposes a model-based approach for system complexity to increase transparency and allow for a visualization of the data-driven analysis in order to address, amongst others, the necessary trade-off between standardization and customer-specific solutions. Yet, none of the discovered publications provides new or different categorizations and metrics besides the ones stated and described above; they all built upon or apply the work and approaches already mentioned.

Third, a strong trend and focus on the economic aspects and implications of complexity and supply chains of the system development has been discovered (Aita Ramírez Gastón, 2020;

Alkan, Vera, Ahmad, & Harrison, 2018; Bouhaddou & Benabdelhafid, 2017; Hidalgo & Hausmann, 2009; Hvam, Hansen, Forza, Mortensen, & Haug, 2020; Maggioni, Lo Turco, & Gallegati, 2016; Mesa, Esparragoza, & Maury, 2018; Modrak & Soltysova, 2017). These efforts target the implications of the system complexity regarding supply chains and how the relations to economic factors can be evaluated. Maggioni, Lo Turco, and Gallegati (2016), for example, relate the complexity of a system to the overall firm/company complexity and the volatility of its outcome. This way, they enable an assessment of the economic impact and how it is related to predictability/fluctuations. While none of these efforts pose any new general categorizations or metrics, some are very well in line with the research in this dissertation. For instance, Alkan et al. (2018) describe their approach to assess and predict assembly complexity of industrial products in the early design phases. While this is not directly the same topic and methodology as the core contributions of this work, it shows that a predictive approach and early consideration are fruitful and applicable, which further supports the research in this dissertation.

Only one new categorization approach overall was found published in the recent past, which was presented by Zhang & Thomson (2018). In their publication from 2018, the authors describe an approach that builds upon the work of Bashir & Thomson (1999) and further extends the function-based metric to include a knowledge aspect. This way, they add to the technical complexity the aspect of integration complexity, which expresses the potential resulting from the combination and interfaces between the different functions. By extending the technical aspects, the authors claim to capture aspects of "connection, element and topology" (Zhang & Thomson, 2018). To utilize their metric, Zhang & Thomson provide cases to estimate the effort and likely duration of system developments. The authors were able to show potential knowledge discrepancies. These discrepancies were also shown to result in integration issues, which is what Zhang & Thomson claim as the contribution of their metric.

Lastly, and most importantly because directly related, the concept of Problem Complexity (Salado & Nilchiani, 2014b) is mentioned and explained here. This concept also provides a measure for the complexity of a system and introduces the underlying problem as a complexity source that gets added to functional, structural, and organizational complexity, see Figure 2.10 below.

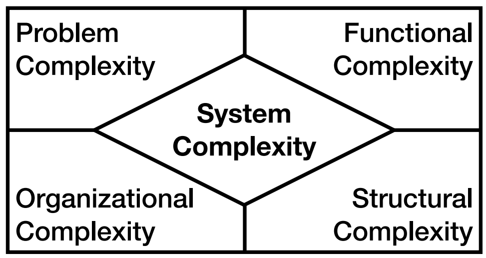

Fig. 2.10 - Complexity framework by Salado & Nilchiani (2014b)

Based on these pillars, Salado & Nilchiani defined the minimum complexity of a system: each of the four types contributes a minimum complexity of the system. To achieve a comparable basis of different complexities herein, the authors utilized the entropy of each type and transformed it into a comparable and summable unit.

The fourth pillar described by Salado & Nilchiani (compared to dimensions in Figure 2.6), the problem complexity, is based on the requirements of a system and therefore the limitations of the solution space. Equation 2.1 shows the formula for problem complexity.

$$C_p = K \cdot \left(\sum_{i=1}^{n} a_i \cdot r_{f_i} \right)^E \cdot \prod_{j=1}^{m} H_j^{\,b_j} \tag{2.1}$$

In this equation, K defines a calibration factor that has to be adjusted based on the assessed case. The variables a and r_f are the weight factor and the number of functional requirements that the system has to fulfill. The weight of each requirement is based on the difficulty it poses. The whole sum in the first term is exponentiated with a diseconomies of scale factor E. The second half of the equation is the product of the influence of the requirement conflicts H exponentiated each with a respective factor b to account for the influence and diseconomies of scale.

The entire term in Equation 2.1 forms the problem complexity C_p to be added to the functional complexity, structural complexity, and organizational complexity. Since problem complexity is one of the four dimensions, it defines the minimum of the overall complexity regardless of the magnitude of the other three. It has to be noted that the problem complexity described by Salado & Nilchiani (2014b) is solely based on functional requirements, influencing factors for them, and conflicts between them. While this allows for an adequate evaluation based on the solution space, it is not necessarily all-embracive as the influence of other requirements, such as performance requirements (Salado & Nilchiani, 2014a), for instance, is omitted. Thus, the concept was considered in the work and contribution of this dissertation.

This last publication marks the end of the literature research concerning system complexity metrics/measures. In conclusion, looking at the complexity characteristics and measures, it becomes clear that a number of them exist based on the different purposes and research fields they originated from. Furthermore, other approaches specifically target projects and processes in their entirety, such as Schuh et al., with a focus on manufacturing (Schuh et al., 2016). Overall, the researched metrics and complexity topics show that work has been done in the intended direction, but at the same time also reveal that the work and purpose of this dissertation to assess requirements in their totality to allow for a predictive and anticipating approach from the beginning onwards has not yet been undertaken or attempted. Therefore, the approaches researched above were considered and assessed as to which aspects can and will be included, but none of the approaches above cover the topic of this dissertation in its entirety.

The last approach mentioned, the problem complexity framework by Salado & Nilchiani (2014b), is the closest related to the research in this dissertation as it addresses a partially similar topic. Yet, the concept of problem complexity is not comprehensive as it only concerns functions and does not allow for any assessment of the complexity of the system in its entirety from the beginning onwards. Thus, while in line and supportive of the research of this dissertation, the work by Salado & Nilchiani (2014b) and still leaves room for expansion and was considered accordingly.

This concludes the state of the research regarding system complexity characteristics and metrics thereof. Based on all the literature assessed in the previous paragraphs and sections, the next two chapters address the remaining two topics evaluated as part of the literature review: Requirements and Natural Language Processing.

Bibliography of Chapter 2

Açıkgöz, A., Günsel, A., Kuzey, C., & Seçgin, G. (2016). Functional Diversity, Absorptive Capability and Product Success: The Moderating Role of Project Complexity in New Product Development Teams. *Creativity and Innovation Management, 25*(1), 90-109. doi:10.1111/caim.12155

Adami, C., Ofria, C., & Collier, T. C. (2000). Evolution of biological complexity. *Proceedings of the National Academy of Sciences, 97*(9), 4463-4468. doi:10.1073/pnas.97.9.4463

Aita Ramírez Gastón, R. (2020). Which complexity characteristics do economical industries present? *International Journal of Design and Nature and Ecodynamics, 15*(2), 177-182. doi:10.18280/ijdne.150206

Albrecht, A. J. (1979, 14-17 October). *Measuring Application Development Productivity.* Paper presented at the Proceedings of the Joint SHARE/GUIDE/IBM Applications Development Symposium.

Alkan, B., Vera, D., Ahmad, B., & Harrison, R. (2018). A Method to Assess Assembly Complexity of Industrial Products in Early Design Phase. *IEEE Access, 6*, 989-999. doi:10.1109/ACCESS.2017.2777406

Baccarini, D. (1996). The concept of project complexity—a review. *International Journal of Project Management, 14*(4), 201-204. doi:10.1016/0263- 7863(95)00093-3

Baldwin, C. Y., & Clark, K. B. (2000). *Design Rules, Vol. 1: The Power of Modularity*. Cumberland, RI: MIT Press.

Bashir, H. A., & Thomson, V. (1999). Estimating Design Complexity. *Journal of Engineering Design, 10*(3), 247-257. doi:10.1080/095448299261317

Bashir, H. A., & Thomson, V. (2004). Estimating design effort for GE hydro projects. *Computers & Industrial Engineering, 46*(2), 195-204. doi:10.1016/j.cie.2003.12.005

Basili, V. R., Briand, L. C., & Melo, W. L. (1996). A validation of object-oriented design metrics as quality indicators. *IEEE Transactions on Software Engineering, 22*(10), 751-761. doi:10.1109/32.544352

Benabdellah, A. C., Bouhaddou, I., & Benghabrit, A. (2019, 22-25 April). *Holonic multi-agent system for modeling complexity structures of Product Development Process.* Paper presented at the 2019 4th World Conference on Complex Systems (WCCS).

Bonchev, D., & Rouvray, D. H. (2005). *Complexity in Chemistry, Biology, and Ecology*. Boston, MA: Springer Science+Business Media.

Bouhaddou, I., & Benabdelhafid, A. (2017). Product Lifecycle Management (PLM): A Key to Manage Supply Chain Complexity. In P. Bourgine, P. Collet, & P. Parrend (Eds.), *First Complex Systems Digital Campus World E-Conference 2015* (pp. 375-382). Cham, Switzerland: Springer International Publishing.

Bozarth, C. C., Warsing, D. P., Flynn, B. B., & Flynn, E. J. (2009). The impact of supply chain complexity on manufacturing plant performance. *Journal of Operations Management, 27*(1), 78-93. doi:10.1016/j.jom.2008.07.003

Braha, D., & Maimon, O. (2013). *A Mathematical Theory of Design: Foundations, Algorithms and Applications*: Springer Science & Business Media.

Brooks, F. P. (1987). No Silver Bullet Essence and Accidents of Software Engineering. *Computer, 20*(4), 10-19. doi:10.1109/MC.1987.1663532

Bryant, P. R. (1959). The order of complexity of electrical networks. *Proc. Inst. Elec. Eng., 106*, 174-188.

Budd, R. W., Thorp, R. K., & Donohew, L. (1967). Content Analysis of Communications.

Carlson, J. M., & Doyle, J. (2002). Complexity and robustness. *Proceedings of the National Academy of Sciences, 99*(suppl 1), 2538-2545. doi:10.1073/pnas.012582499

Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering, 20*(6), 476-493. doi:10.1109/32.295895

Cilliers, P. (2000). What Can We Learn From a Theory of Complexity? *EMERGENCE, 2*(1), 23-33. doi:10.1207/S15327000EM0201_03

Collberg, C., Thomborson, C., & Low, D. (1997). *A Taxonomy of Obfuscating Transformations*.

Corning, P. A. (1998). Complexity is just a word! *Technological Forecasting and Social Change, 59*(2), 197-200.

Cotler, J., Hunter-Jones, N., Liu, J., & Yoshida, B. (2017). Chaos, complexity, and random matrices. *Journal of High Energy Physics, 2017*(11), 48. doi:10.1007/JHEP11(2017)048

Cotsaftis, M. (2009). What Makes a System Complex? - An Approach to Self Organization and Emergence. In M. A. Aziz-Alaoui & C. Bertelle (Eds.), *From System Complexity to Emergent Properties* (pp. 49-99). Berlin, Heidelberg: Springer Berlin Heidelberg.

Crawley, E., Cameron, B., & Selva, D. (2015). *System Architecture: Strategy and Product Development for Complex Systems*: Prentice Hall Press.

Crutchfield, J. P., & Wiesner, K. (2010). Simplicity and Complexity. *Physics World, February 2010*, 36-38.

Cui, A. S., & Wu, F. (2017). The Impact of Customer Involvement on New Product Development: Contingent and Substitutive Effects. *Journal of Product Innovation Management, 34*(1), 60-80. doi:10.1111/jpim.12326

Dandy, G., Walker, D., Daniell, T., & Warner, R. (2008). *Planning and Design of Engineering Systems* (second ed.): Taylor & Francis.

DeRosa, J. K., Grisogono, A., Ryan, A. J., & Norman, D. O. (2008, 7-10 April). *A Research Agenda for the Engineering of Complex Systems.* Paper presented at the 2008 2nd Annual IEEE Systems Conference, Montreal, Canada.

Dierneder, S., & Scheidl, R. (2001, 19-23 February). *Complexity Analysis of Systems from a Functional and Technical Viewpoint.* Paper presented at the International Conference on Computer Aided Systems Theory - EUROCAST, Las Palmas de Gran Canaria, Spain.

ElMaraghy, W., ElMaraghy, H., Tomiyama, T., & Monostori, L. (2012). Complexity in engineering design and manufacturing. *CIRP Annals, 61*(2), 793-814. doi:10.1016/j.cirp.2012.05.001

Eun Sook, C., Min Sun, K., & Soo Dong, K. (2001, 4-7 December). *Component metrics to measure component quality.* Paper presented at the Proceedings Eighth Asia-Pacific Software Engineering Conference.

Fain, N., Žavbi, R., & Vukašinović, N. (2016, May 16-19). *The influence of product complexity on team performance within NPD.* Paper presented at the Proceedings of the DESIGN 2016 14th International Design Conference, Dubrovnik, Croatia.

Fenton, N., & Bieman, J. (2014). *Software Metrics: A Rigorous and Practical Approach, Third Edition*. Boca Raton, FL: CRC Press.

Frodeman, R., Klein, J. T., & Dos Santos Pacheco, R. C. (2017). *The Oxford Handbook of Interdisciplinarity* (second ed.): Oxford University Press.

Frodeman, R., Klein, J. T., Mitcham, C., & Holbrook, J. B. (2010). *The Oxford Handbook of Interdisciplinarity* (first ed.): Oxford University Press.

Gell-Mann, M., & Lloyd, S. (1996). Information measures, effective complexity, and total information. *Complexity, 2*(1), 44-52. doi:10.1002/(SICI)1099-0526(199609/10)2:1<44::AID-CPLX10>3.0.CO;2-X

Ghosh, A., Kristjandottir, K., Hvam, L., & Vareilles, É. (2018, 2018-09-27). *Measuring the Complexity of Product Configuration Systems.* Paper presented at the ConfWS 2018 - 20th Configuration Workshop, Graz, Austria.

Gonzalez Castro, S., Panarotto, M., Borgue, O., & Isaksson, O. (2020, 12-14 August). *Analysing increase of functionality and complexity in integrated product architectures.* Paper presented at the NordDesign 2020, Lyngby, Denmark.

Göpfert, J. (1998). Modulare Produktentwicklung. In N. Franke & C.-F. von Braun (Eds.), *Innovationsforschung und Technologiemanagement: Konzepte, Strategien, Fallbeispiele* (pp. 139-151). Berlin Heidelberg, Germany: Springer-Verlag.

Griffin, A. (1993). Metrics for Measuring Product Development Cycle Time. *Journal of Product Innovation Management, 10*(2), 112-125. doi:10.1111/1540-5885.1020112

Griffin, A. (1997a). The Effect of Project and Process Characteristics on Product Development Cycle Time. *Journal of Marketing Research, 34*(1), 24-35. doi:10.2307/3152062

Griffin, A. (1997b). Modeling and measuring product development cycle time across industries. *Journal of Engineering and Technology Management, 14*(1), 1-24. doi:10.1016/S0923-4748(97)00004-0

Gupta, S., & Krishnan, V. (1999). Integrated component and supplier selection for a product family. *Production and Operations Management, 8*(2), 163-182. doi:10.1111/j.1937-5956.1999.tb00368.x

Halstead, M. H. (1977). *Elements of Software Science*: Elsevier.

Hamilton, J. (2000). *The Engineering Profession*. Retrieved from

Hassan, A. E. (2009, 16-24 May 2009). *Predicting faults using the complexity of code changes.* Paper presented at the 2009 IEEE 31st International Conference on Software Engineering.

Hazen, R. M., Griffin, P. L., Carothers, J. M., & Szostak, J. W. (2007). Functional information and the emergence of biocomplexity. *Proceedings of the National Academy of Sciences, 104*(suppl 1), 8574-8581. doi:10.1073/pnas.0701744104

Henry, S., & Kafura, D. (1981). Software Structure Metrics Based on Information Flow. *IEEE Transactions on Software Engineering, SE-7*(5), 510-518. doi:10.1109/TSE.1981.231113

Heylighen, F. (1999). The Growth of Structural and Functional Complexity during Evolution. In F. Heylighen, J. Bollen, & A. Riegler (Eds.), *The Evolution of Complexity* (pp. 17-44). Dordrecht, Netherlands: Kluwer Academic Publishers.

Hidalgo, C. A., & Hausmann, R. (2009). The building blocks of economic complexity. *Proceedings of the National Academy of Sciences, 106*(26), 10570-10575. doi:10.1073/pnas.0900943106

Holland, J. H. (1996). *Hidden Order*. Reading, MA: Addison-Wesley.

Hvam, L., Hansen, C. L., Forza, C., Mortensen, N. H., & Haug, A. (2020). The reduction of product and process complexity based on the quantification of product complexity costs. *International Journal of Production Research, 58*(2), 350-366. doi:10.1080/00207543.2019.1587188

INCOSE. (2019). Systems Engineering - Transforming Needs to Solutions. Retrieved from https://www.incose.org/systems-engineering

INCOSE. (2023). Careers in SE. Retrieved from https://www.incose.org/about-systems-engineering/careers-in-se

Isik, F. (2010). An entropy-based approach for measuring complexity in supply chains. *International Journal of Production Research, 48*(12), 3681-3696. doi:10.1080/00207540902810593

Jacobs, M. A. (2007). Product Complexity: A Definition and Impacts on Operations. *Decision Line, 38*(5), 6-12.

Jacobs, M. A., & Swink, M. (2011). Product portfolio architectural complexity and operational performance: Incorporating the roles of learning and fixed assets. *Journal of Operations Management, 29*(7), 677-691. doi:10.1016/j.jom.2011.03.002

Kannapan, S. M. (1995). Function Metrics for Engineered Devices. *Applied Artificial Intelligence, 9*(1), 45-64. doi:10.1080/08839519508945467

Katsikeas, C. S., Leonidou, C. N., & Zeriti, A. (2016). Eco-friendly product development strategy: antecedents, outcomes, and contingent effects. *Journal of the Academy of Marketing Science, 44*(6), 660-684. doi:10.1007/s11747-015-0470-5

Kauffman, S. A. (1996). *At Home in the Universe*. Oxford, United Kingdom: Oxford University Press.

Kim, G., Kwon, Y., Suh, E. S., & Ahn, J. (2016). Analysis of Architectural Complexity for Product Family and Platform. *Journal of Mechanical Design, 138*(7). doi:10.1115/1.4033504

Kim, S., & Moon, S. K. (2019). Eco-modular product architecture identification and assessment for product recovery. *Journal of Intelligent Manufacturing, 30*(1), 383-403. doi:10.1007/s10845-016-1253-7

Kossiakoff, A., Sweet, W. N., Seymour, S. J., & Biemer, S. M. (2011). *Systems Engineering Principles and Practice* (second ed.): John Wiley & Sons.

Lindemann, U., Maurer, M., & Bran, T. (2009). *Structural Complexity Management*. Berlin Heidelberg, Germany: Springer-Verlag.

Lloyd, S., & Pagels, H. (1988). Complexity as thermodynamic depth. *Annals of Physics, 188*(1), 186-213. doi:10.1016/0003-4916(88)90094-2

Maggioni, D., Lo Turco, A., & Gallegati, M. (2016). Does product complexity matter for firms' output volatility? *Journal of Development Economics, 121*, 94-109. doi:10.1016/j.jdeveco.2016.03.006

Malmiry, R. B., Dantan, J.-Y., Pailhès, J., & Antoine, J.-F. (2016). A product functional modelling approach based on the energy flow by using characteristics-properties modelling. *Journal of Engineering Design, 27*(12), 817-843. doi:10.1080/09544828.2016.1235261

Malmiry, R. B., Pailhès, J., Qureshi, A. J., Antoine, J.-F., & Dantan, J.-Y. (2016). Management of product design complexity due to epistemic uncertainty via energy flow modelling based on CPM. *CIRP Annals, 65*(1), 169-172. doi:10.1016/j.cirp.2016.04.048

Mar, B. W. (1997). BACK TO BASICS AGAIN – A SCIENTIFIC DEFINITION OF SYSTEMS ENGINEERING. *INCOSE International Symposium, 7*(1), 309-316. doi:10.1002/j.2334-5837.1997.tb02187.x

Mauerhoefer, T., Strese, S., & Brettel, M. (2017). The Impact of Information Technology on New Product Development Performance. *Journal of Product Innovation Management, 34*(6), 719-738. doi:10.1111/jpim.12408

McCabe, T. J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering, SE-2*(4), 308-320. doi:10.1109/TSE.1976.233837

McShea, D. W. (2001). The hierarchical structure of organisms: a scale and documentation of a trend in the maximum. *Paleobiology, 27*(2), 405-423. doi:10.1666/0094-8373(2001)027<0405:Thsooa>2.0.Co;2

The Merriam-Webster Dictionary, International Edition. (2016). Merriam-Webster Inc.

Mesa, J. A., Esparragoza, I., & Maury, H. (2018). Development of a metric to assess the complexity of assembly/disassembly tasks in open architecture products. *International Journal of Production Research, 56*(24). doi:10.1080/00207543.2017.1398431

Modrak, V., & Soltysova, Z. (2017). Novel Complexity Indicator of Manufacturing Process Chains and Its Relations to Indirect Complexity Indicators. *Complexity*. doi:10.1155/2017/9102824

Mourtzis, D., Fotia, S., & Boli, N. (2017, 27-29 June). *Metrics definition for the product-service system complexity within mass customization and industry 4.0 environment.* Paper presented at the 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC).

Orfi, N., Terpenny, J., & Sahin-Sariisik, A. (2011). Harnessing Product Complexity: Step 1—Establishing Product Complexity Dimensions and Indicators. *The Engineering Economist, 56*(1), 59-79. doi:10.1080/0013791X.2010.549935

Orfi, N., Terpenny, J., & Sahin-Sariisik, A. (2012). Harnessing Product Complexity: Step 2—Measuring and Evaluating Complexity Levels. *The Engineering Economist, 57*(3), 178-191. doi:10.1080/0013791X.2012.702197

Oviedo, E. I. (1980, October 27-31). *Control Flow, Data Flow and Program Complexity.* Paper presented at the Proceedings of the 4th Computer Software and Applications Conference, COMPSAC, Chicago, IL.

Panos, C. P., Chatzisavvas, K. C., Moustakidis, C. C., Nikolaidis, N., Massen, S. E., & Sen, K. D. (2011). Atomic Statistical Complexity. In K. D. Sen (Ed.), *Statistical Complexity: Applications in Electronic Structure* (pp. 49-64). Dordrecht: Springer Netherlands.

Park, K., & Okudan Kremer, G. E. (2019). An investigation on the network topology of an evolving product family structure and its robustness and complexity. *Research in Engineering Design, 30*(3), 381-404. doi:10.1007/s00163-019-00310-y

Phelan, S. E. (2001). What Is Complexity Science, Really? *EMERGENCE, 3*(1), 120-136. doi:10.1207/S15327000EM0301_08

Portugali, J., Meyer, H., Stolk, E., & Tan, E. *Complexity Theories of Cities Have Come of Age*. Berlin Heidelberg, Germany: Springer-Verlag.

Raja, V., Kokkolaras, M., & Isaksson, O. (2019). A simulation-assisted complexity metric for design optimization of integrated architecture aero-engine structures. *Structural and Multidisciplinary Optimization, 60*(1), 287-300. doi:10.1007/s00158-019-02308-5

Read, C. (2008). *Complexity Characteristics Measurement Engineering System.* (Doctor of Philosophy). Loughborough University, Loughborough, United Kingdom.

Richardson, K. A., Cilliers, P., & Lissack, M. (2001). Complexity Science: A "Gray" Science for the "Stuff in Between". *EMERGENCE, 3*(2), 6-18.

Romero, P., Obradovic, Z., Li, X., Garner, E. C., Brown, C. J., & Dunker, A. K. (2001). Sequence complexity of disordered protein. *Proteins: Structure, Function, and Bioinformatics, 42*(1), 38-48. doi:10.1002/1097-0134(20010101)42:1<38::AID-PROT50>3.0.CO;2-3

Salado, A., & Nilchiani, R. (2014a). A Categorization Model of Requirements Based on Max-Neef's Model of Human Needs. *Systems Engineering, 17*(3), 348-360. doi:10.1002/sys.21274

Salado, A., & Nilchiani, R. (2014b). The Concept of Problem Complexity. *Procedia Computer Science, 28*, 539-546. doi:10.1016/j.procs.2014.03.066

Sanders, I. (2003). *WHAT IS COMPLEXITY?* Retrieved from Washington, DC:

Schöggl, J.-P., Baumgartner, R. J., & Hofer, D. (2017). Improving sustainability performance in early phases of product design: A checklist for sustainable product development tested in the automotive industry. *Journal of Cleaner Production, 140*, 1602-1617. doi:10.1016/j.jclepro.2016.09.195

Schuh, G., Dölle, C., & Koch, J. (2018, 28 June-1 July). *Data Model-based Improvement of Transparency and Controllability of Product Complexity.* Paper presented at the 2018 IEEE Technology and Engineering Management Conference (TEMSCON).

Schuh, G., Rudolf, S., & Mattern, C. (2016, 14-17 March). *Conceptual framework for evaluation of complexity in new product development projects.* Paper presented at the 2016 IEEE International Conference on Industrial Technology (ICIT).

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*(3), 379-423. doi:10.1002/j.1538-7305.1948.tb01338.x

Sheard, S., & Mostashari, A. (2011). Complexity in large-scale technical project management. *International Journal of Complexity in Leadership and Management, 1*(3), 289-300. doi:10.1504/ijclm.2011.042550

Sheard, S. A., & Mostashari, A. (2009). Principles of complex systems for systems engineering. *Systems Engineering, 12*(4), 295-311. doi:10.1002/sys.20124

Sihvonen, A., & Pajunen, K. (2019). Causal complexity of new product development processes: a mechanism-based approach. *Innovation, 21*(2), 253-273. doi:10.1080/14479338.2018.1513333

Simon, H. A. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society, 106*(6), 467-482. Retrieved from http://www.jstor.org/stable/985254

Sinha, K. (2014). *Structural Complexity and its Implications for Design of Cyber-Physical Systems.* (Doctor of Philosophy). Massachusetts Institute of Technology,

Sinha, K., & de Weck, O. L. (2013, 15-18 April 2013). *A network-based structural complexity metric for engineered complex systems.* Paper presented at the 2013 IEEE International Systems Conference (SysCon).

Sinha, K., Han, S.-Y., & Suh, E. S. (2020). Design structure matrix-based modularization approach for complex systems with multiple design constraints. *Systems Engineering, 23*(2), 211-220. doi:10.1002/sys.21518

Sinha, K., & Suh, E. S. (2018). Pareto-optimization of complex system architecture for structural complexity and modularity. *Research in Engineering Design, 29*(1), 123-141. doi:10.1007/s00163-017-0260-9

Sinha, K., Suh, E. S., & de Weck, O. (2018). Integrative Complexity: An Alternative Measure for System Modularity. *Journal of Mechanical Design, 140*(5). doi:10.1115/1.4039119

Snowden, D. J., & Boone, M. E. (2007). A leader's framework for decision making. *Harvard business review, 85*(11), 68.

Strogatz, S. H. (2001). Exploring complex networks. *Nature, 410*(6825), 268-276. doi:10.1038/35065725

Strogatz, S. H. (2003). *SYNC: the emerging science of spontaneous order*. New York, NY: Hyperion.

Strogatz, S. H. (2012). *Sync: How Order Emerges from Chaos in the Universe, Nature, and Daily Life* (1st ed.). New York, NY: Hyperion.

Subramanyam, R., & Krishnan, M. S. (2003). Empirical analysis of CK metrics for object-oriented design complexity: implications for software defects. *IEEE Transactions on Software Engineering, 29*(4), 297-310. doi:10.1109/TSE.2003.1191795

Suh, N. P. (1999). A Theory of Complexity, Periodicity and the Design Axioms. *Research in Engineering Design, 11*(2), 116-132. doi:10.1007/PL00003883

Suh, N. P. (2005). *Complexity: theory and applications*: Oxford university press.

Summers, J. D., & Shah, J. J. (2010). Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability. *Journal of Mechanical Design, 132*(2). doi:10.1115/1.4000759

Systems Engineering Across Multiple Domains. (2014). Retrieved from https://ep.jhu.edu/about-us/news-and-media/systems-engineering-across-multiple-domains

Turner, J. R., & Cochrane, R. A. (1993). Goals-and-methods matrix: coping with projects with ill defined goals and/or methods of achieving them. *International Journal of Project Management, 11*(2), 93-102. doi:10.1016/0263-7863(93)90017-H

Vandergriff, L. J. (2007, 11-12 January). *System Engineering in the 21st Century - Implications from Complexity.* Paper presented at the Symposium on Complex Systems Engineering, RAND Corporation, Santa Monica, CA.

Vierlboeck, M., & Nilchiani, R. R. (2021, 13 Sept.-13 Oct. 2021). *Requirement Engineering in the Age of System and Product Complexity – A Literature Review.* Paper presented at the 2021 IEEE International Symposium on Systems Engineering (ISSE).

Vogel, W., & Lasch, R. (2018). Complexity drivers in product development: A comparison between literature and empirical research. *Logistics Research, 11*(7), 1-42. doi:10.23773/2018_7

von Bary, N., Rebentisch, E., & Becerril, L. (2018). *Simulation-based Value Analysis of Organizational Complexity in Product Development Projects.* Paper presented at the 20th International Dependency and Structure Modeling Conference, Trieste, Italy.

Vyron, D., Panos, F., & James, F. O. K. (2018). Modeling Software Development Process Complexity. *International Journal of Information Technology Project Management (IJITPM), 9*(4), 17-40. doi:10.4018/IJITPM.2018100102

Weaver, W. (1948). Science and Complexity. *American Scientist, 36*, 536-544.

Weber, C. (2005, 15-18 August). *What is "Complexity"?* Paper presented at the ICED 05: 15th International Conference on Engineering Design, Melbourne, Australia.

Weyuker, E. J. (1988). Evaluating software complexity measures. *IEEE Transactions on Software Engineering, 14*(9), 1357-1365. doi:10.1109/32.6178

Xiao, Z., Zhou, Z., & Sheng, B. (2016). A New Classification Analysis of Customer Requirement Information Based on Quantitative Standardization for Product Configuration. *Mathematical Problems in Engineering, 2016*. doi:10.1155/2016/7274538

Yang, Q., Shan, C., Jiang, B., Yang, N., & Yao, T. (2018). Managing the complexity of new product development project from the perspectives of customer needs and entropy. *Concurrent Engineering, 26*(4), 328-340. doi:10.1177/1063293x18798001

Zhang, X., & Thomson, V. (2018). A knowledge-based measure of product complexity. *Computers & Industrial Engineering, 115*, 80-87. doi:10.1016/j.cie.2017.11.005

CHAPTER 3: REQUIREMENTS ENGINEERING

"Plans are worthless, but planning is everything".

Dwight D. Eisenhower

With the information and overview regarding complexity provided in Chapter 2, the other two pillars of the presented work can be addressed. Thus, Chapter 3 looks at the history of requirements (Section 3.1) as well as the structural aspects and complexity-related topics (Section 3.2). In the general and history section, all information is provided in chronological order, corresponding to the way it was researched.

3.1 General Aspects and History *

When it comes to requirements and requirements engineering (RE) in a general manner, scientific publications can be found addressing the topic as early as the 1960s (Dresner & Borchers, 1964) in the Software Engineering field (Callele, Wnuk, & Penzenstadler, 2017). The term requirements goes back even further, all the way into the 19th century (Kolligs & Thomas, 2020), and through time, various publications that are significant in the field of RE have to be mentioned. An overview is provided, including the fields of the publications.

As mentioned above, the management of requirements and also the term RE can be traced back to the 1960s (Dresner & Borchers, 1964) when the first military standards considered close to RE were developed (Department of Defense, 1969). Despite these occurrences, the widespread use of the term requirements engineering as a professional discipline and field did not emerge until the end of the twentieth century (Mead, 2013). This is further underlined by the scarcity of publications before the last decade of the twentieth century.

* The content of this section has been extracted literally or with minor editorial modifications from M. Vierlboeck and R. Nilchiani (2022), "Requirement Engineering in the Age of System and Product Complexity – A Literature Review," published by IEEE.

For instance, limiting the search timeframe to the area before the standards and widely adopted approaches (described in the paragraph below), leaves only a small number of publications with more than 25 citations (as of November 2022) that can be considered established and popular. These publications are outlined hereinafter.

The first important contribution comes from Alford, who, together with various authors, in 1977 developed the Software Requirements Engineering Methodology (SREM) (Alford, 1977). SREM was designed for software and weapons systems and addresses activities ranging from generation to the validation of requirements. The methodology relies on system functions already being allocated to the respective processor and data being collected. Overall, the approach can be seen as comprehensive since it covers various aspects of the requirement management and includes steps all the way to validation. Yet, the approach was and is still geared towards software and, as shown by the reliance and assumptions, limited. Alford further evolved the approach throughout the years (Alford, 1978, 1985).

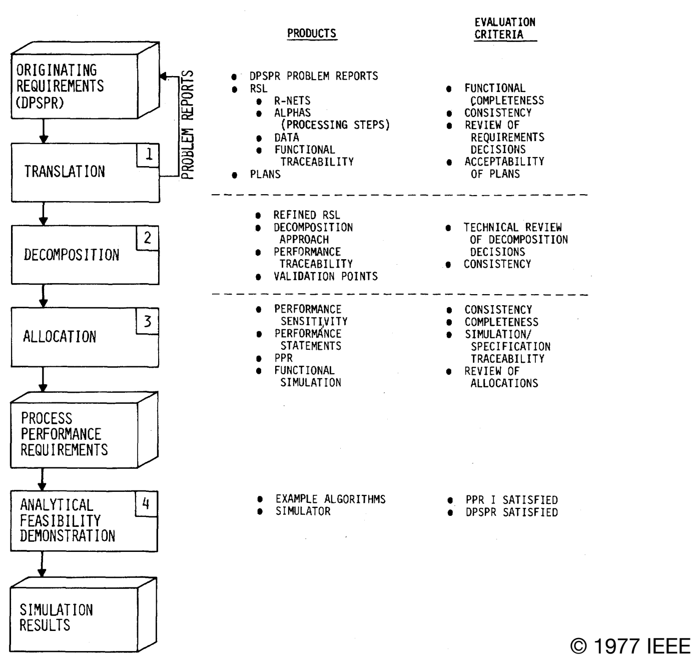

FIG. 3.1 - SREM METHODOLOGY OVERVIEW AND STEPS (ALFORD, 1977)

Between the RE work of Alford and the emergence of the first major standards in the 1990s, only remotely related publications were discovered. For example, IEEE published a "Standard Glossary of Software Engineering Terminology" ("IEEE Standard Glossary of Software Engineering Terminology," 1983) in 1983 that included requirements. In addition, for comprehensiveness, the argument can be made that various other approaches, some of them mentioned in the previous chapters, can be considered RE methodologies, despite not being titled or associated with RE at their inception. Such approaches could be the Waterfall Model by Boehm, which is further underlined by various other publications of Boehm addressing requirements and their management (Boehm, 1984). Overall though, until the standards described in the next paragraph were established, no major publications related to the topic of RE were found. Thus, we begin with the explanation of these standards that also coincided with the inception of related journals, such as the Requirements Engineering Journal (Loucopoulos & Mylopoulos, 2003).

Towards the end of the twentieth century, standards by reputable and popular institutions started emerging (IEEE, 1998a, 1998b). Furthermore, in 2011, a conglomerate international standard conflated various previously existing approaches into one general standard (ISO/IEC/IEEE, 2011). In this standard, requirements engineering is defined as follows: "Requirements engineering is an interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system, software or service of interest […]". Furthermore, RE was described as supposed to facilitate an agreement and understanding of the stakeholders and provide a verification basis for design solutions (ISO/IEC/IEEE, 2011). Also, requirements specifically were defined as statements "translate[ing] or express[ing] a need and its associated constraints and conditions" (ISO/IEC/IEEE, 2011). These needs and associating criteria stem from the various stakeholders of the project/product. In addition, the exact acquisition, derivation, and even formulation, as well as phrasing of the requirements, have to comply with a set of characteristics laid out by the standard. Overall, the standard provides a uniform framework for RE and also includes organizational process details.

The emergence of standards marks the beginning of the expansion of RE into areas beyond the software fields. As a result, more and more publications and research started appearing in various scientific and business fields. In order to provide an overview, the following paragraphs include the most important publications since the emergence of the first standards and, as the current state of the research is also critical, the most recent publications and advances in the field of RE.

As described, until the end of the twentieth century, RE was focused on Software Engineering and the related disciplines. Concurrently with the inception of the standards as well as conferences such as the "International Requirements Engineering Conference" by IEEE (Mead, 2013), RE expanded and was taken into consideration in other areas and business fields. Such fields included engineering design in a general sense (Darlington & Culley, 2002; Hsu & Woon, 1998), mechanical engineering (Weber & Weisbrod, 2003), and management (Hales & Gooch, 2004). With this expansion, RE became more widely adopted and can today be found in various fields where systems are being developed, for instance, car design (Ponn & Lindemann, 2011). Moreover, RE has been adopted by guidelines and frameworks, e.g., the NASA Systems Engineering Handbook (National Aeronautics and Space Administration (NASA), 2020).

Despite the expansion and widespread application of RE, the underlying processes and structures are still shared by most fields and have only been adapted where necessary due to different circumstances of the respective are. The expansion of RE has exposed the concepts and processes to various other topics/circumstances and as of the time of this writing (October, 2022), RE is still evolving and being applied/evaluated in new fields. To outline the most recent directions, the next part takes a look at the most recent publications.

Regarding the most current research in the domain of RE, also taking into account its history of over 50 years, numerous publications of the last six years address issues with agile development and requirements engineering (Curcio, Navarro, Malucelli, & Reinehr, 2018; Dalpiaz & Brinkkemper, 2018; Elghariani & Kama, 2016; Franch et al., 2017; Gomes De Oliveira Neto, Horkoff, Knauss, Kasauli, & Liebel, 2017; Rashidah Kasauli, Knauss, Horkoff, Liebel, & de Oliveira Neto, 2021; R. Kasauli, Liebel, Knauss, Gopakumar, & Kanagwa, 2017; Ramadan & Megahed, 2016; Schön, Sedeño, Mejías, Thomaschewski, & Escalona, 2019; Schön, Thomaschewski, & Escalona, 2017; Schön, Winter, Escalona, & Thomaschewski, 2017; Villamizar, Kalinowski, Viana, & Fernández, 2018; Wagner, Méndez Fernández, Kalinowski, & Felderer, 2018; Zamudio, Aguilar, Tripp, & Misra, 2017). Further trends can be seen in the research of the application of data analyses and other algorithms/processing tools to support the requirements engineering processes (Abad, Noaeen, & Ruhe, 2016; AlZu'bi, Hawashin, EIBes, & Al-Ayyoub, 2018; Dalpiaz, Ferrari, Franch, & Palomares, 2018; Franch et al., 2017; Ghasemi, 2018; Khan, Liu, Wen, & Ali, 2019; Maalej, Nayebi, & Ruhe, 2019) as well as an upcoming focus on security of systems/software and resilience (Bulusu, Laborde, Samer Wazan, Barrère, & Benzekri, 2017; Bulusu, Laborde, Wazan, Barrère, & Benzekri, 2018; Dalpiaz, Paja, & Giorgini, 2016; Martins & Gorschek, 2020; Mufti, Niazi, Alshayeb, & Mahmood, 2018; Niazi, Saeed, Alshayeb, Mahmood, & Zafar, 2020; ur Rehman & Gruhn, 2017). Lastly, the comprehensive work by Wagner et al. should be mentioned here, who recently conducted an extensive and international survey regarding the current situation of the application of requirements engineering (Wagner et al., 2019).

To sum up the overall trajectory of the research regarding RE, Figure 3.2 shows a timeline with the mentioned publications as well as an overview of the areas and foci.

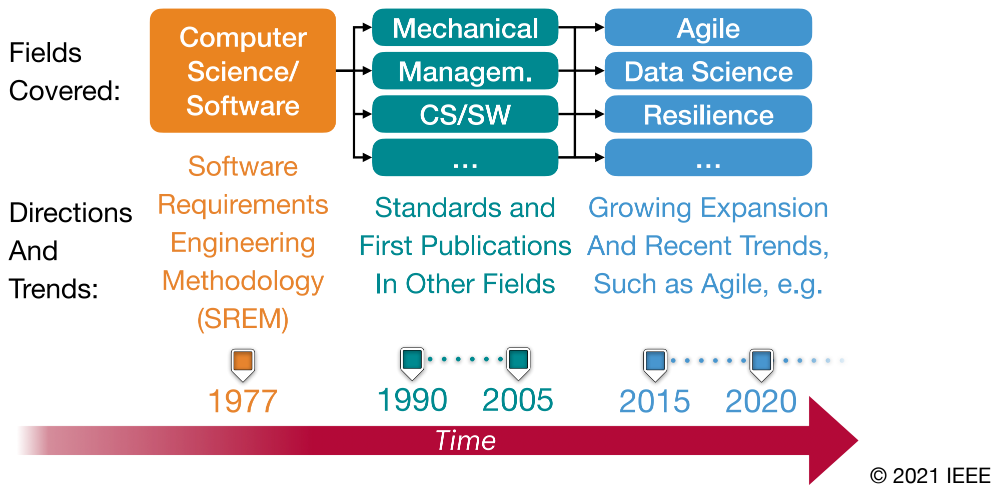

Fig. 3.2 - Timeline and Overview of the RE Research History (Vierlboeck & Nilchiani, 2021)

When it comes to requirements engineering directly in connection with complexity or system complexity, as also described in more detail in the next section, there have been no recent trends or publications in the last five years. Overall, the relevant publications are scarce as most topics situated in the requirements engineering domain address complexity as a phenomenon that needs to be managed (Lindemann, Maurer, & Bran, 2009; Weidmann et al., 2016). Yet, no found publications assessed requirements as a possible source of or interface to system complexity. The closest work regarding these two topics was published by Garina et al. (2021), which addresses the creation of a systems/product development management strategy that considered complexity, including the requirements phase, but does not address nor consider specific complexity aspects therein.

3.2 Requirement Structure and Complexity

Since one of the steps necessary for the research of this dissertation is the categorization and structure definition of requirements, current approaches in this domain were also evaluated. This includes aspects in close relation to the NLP4RE section in Chapter 4.

Requirements can be many-fold and sometimes serve different purposes. For example, achieving completeness in the elicitation process is one goal, albeit "seldom, if ever, achieved"; project planning, risk management, change control (Dick, Hull, & Jackson, 2017), problem definition, and activity planning in the development process (Jain, Chandrasekaran, Elias, & Cloutier, 2008) are additional purposes. Thus, requirement categorization can be defined according to the nature and circumstances of the development. In the following paragraphs, the most popular and widely used categorizations are described to provide a comprehensive overview. Since there are official standards and guidelines by reputable organizations, the approaches are outlined before addressing the classifications found in scientific research. This way, both sides can be evaluated with a clear separation.

First, NASA (2020) describes several categories of requirements, which are primarily geared toward space systems, but applicable to systems engineering and development in general as well. Figure 3.3 on the next page shows the types outlined by the Handbook (National Aeronautics and Space Administration (NASA), 2020).

As shown by the dots in Figure 3.3, some requirement subcategories are variable and can even depend on the individual project at hand. In addition to the distinctions shown, NASA assigns the requirements to two broader categories: functional needs requirements plus their drivers and cross-system levied requirements. The difference herein is that the first category is directly assigned to design features and therefore pertains to the technical requirements, which are driven by the operational requirements and affected by Specialty Requirements. Reliability

Requirements and Safety Requirements are levied across systems as they cannot be assigned directly to parts, components, or elements and can include more than one process/project.

FIG. 3.3 - NASA REQUIREMENT CATEGORIES (EXCERPT FROM THE NASA SYSTEMS ENGINEERING HANDBOOK (2020))

Similar to NASA, the ECSS (European Cooperation for Space Standardization)(2009) provides classifications for requirements: functional, mission, interface, environmental, operational, human factor logistics support, physical, product assurance, configuration, design, and verification requirements.

Third, the ISO/IEC/IEEE Standard 29148 describes requirements as well with the purpose to enable an agreed understanding, allow for validated implementation, and provide a basis for verification (ISO/IEC/IEEE, 2011). For this purpose, the standard distinguishes between the following types and categories of requirements: functional, performance (encompassing usability/quality), interface, process, and non-functional requirements in combination with design constraints. The herein set requirements categories, except the non-functional ones, are defined with specific aspects of the system in mind and also outline how statements should be phrased as well as organized.

Fourth, INCOSE (the International Council of Systems Engineering) proposes a classification for requirements in the Systems Engineering Handbook (National Aeronautics and Space Administration (NASA), 2020), cross-referencing the above-described ISO/IEC/IEEE International Standard 29148 (ISO/IEC/IEEE, 2011). In the handbook, INCOSE describes requirements with to the classification below: functional, performance, usability, interface, operational, modes and/or states, and adaptability requirements. An approach similar to the INCOSE categorization can be found in Dick, Hull, and Jackson's book (2017), which is applicable to Software Engineering.

The four above-mentioned approaches all provide a very clear categorization when it comes to the functional requirement types and leave a lot of freedom as far as the non-functional requirements are concerned. This makes an overall categorization more difficult but also confines the lack of uniformity to a specific part of the requirements. Still, a general and shared frame is not necessarily provided as these standards and rules, despite aiming for universality, have to be applicable to a plethora of situations and circumstances. Therefore, they cannot be excessively restrictive, as this would limit their applicability. Thus, in order to find/

define a general structure, the need for a wider field of view arises beyond the standards and rules. This is described in the following paragraphs.

The first interpretation and approach regarding system requirements that is not a standard or guideline was provided in 1993 by Wymore (1993). In the book about model-based systems engineering, the author proposes a requirement classification not based on projects and or fields but simply based on developed products and systems which serve the purpose of solving a problem. Thus, the author defines the following types of requirements: input/output requirements, technology requirements, performance requirements, cost requirements, trade-off requirements, and system test requirements. This categorization of requirements is supposed to help with the development of the system design and prevent the implementation and restrictions that could result from preconceived solutions and ideas. Therefore, Wymore tried to disconnect the requirements categorization from the circumstances and or field of work.

Directly extending the approach by Wymore, Buede (2009) built upon the results and organized them in accordance with the system life cycle. This led to a hierarchical structure of the above-mentioned requirements and enabled the association of certain requirement types with elements of the development cycle. The final organization and structure presented by Buede are depicted in Figure 3.4 and explained thereinafter.

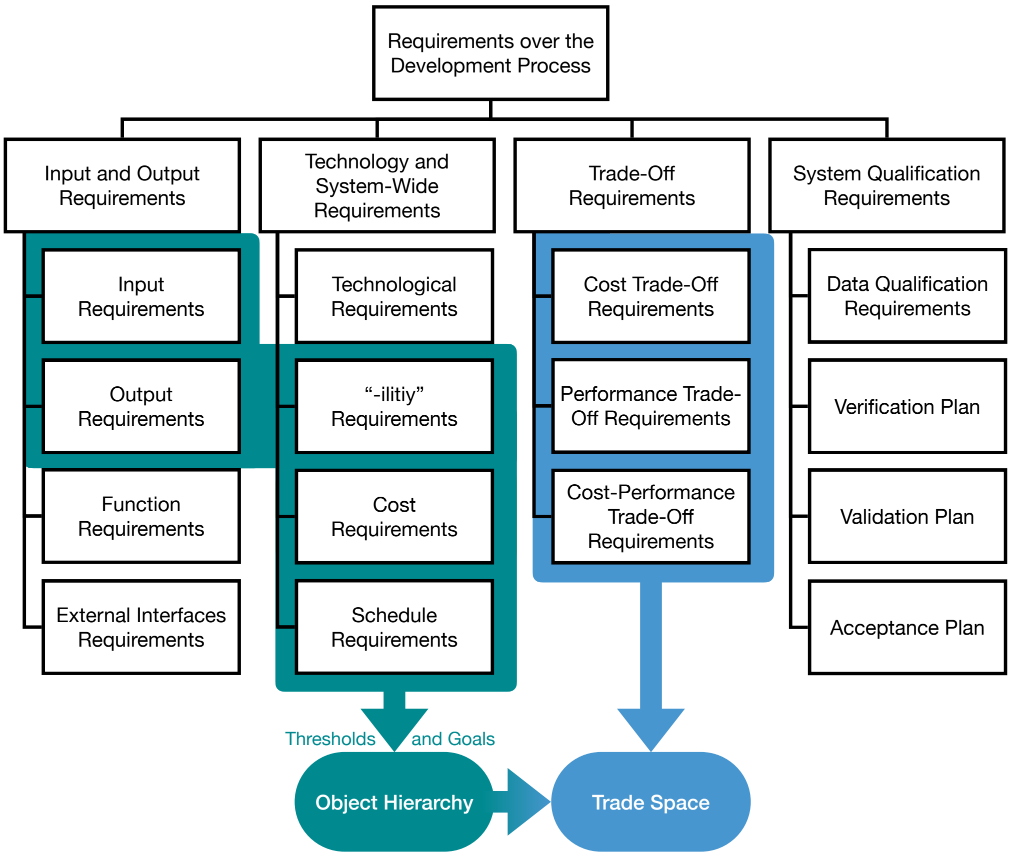

Fig. 3.4 - Requirements across the Development Process (Buede, 2009)

As the figure shows, the hierarchy and structure developed by Buede (2009) assign and associates certain aspects and parts of the requirements with the actual development cycle points as well as design aspects specifically. The highlighted parts of Figure 3.4 indicate that the Input, Output, "-ility," cost, and schedule requirements directly relate to the resulting object hierarchy via the set thresholds and goals. This results in a clear importance hierarchy of the derived objects, which again influences the trade space. The trade space is derived from the Trade-Off Requirements, which are not necessarily all individual requirements per se, but the relations and value connections of various aspects. Therefore, these trade-offs shape the trade space as they require potential compromises in order to be fulfilled. Overall, Buede presented a structured and organized way of categorizing requirements without pertaining to a certain way and or type of system development. The causality linking the requirements to the trade space and object hierarchy shows a clear connection between the requirements and the design/elements and is important to consider for the work at hand.

The next categorization approach was provided by Gilb in 1997 (1997). Herein, the author maps requirements into four categories: Functions, Qualities, Costs, and Constraints. Within these categories, Gilb also differentiates between not-quantifiable (yet testable if present) and quantifiable (measuring scale existing) types. It is important to note here that Gilb defined requirements in a very broad sense, which limits the number of restrictions and conditions posed by the categorization. For example, functions in the context of Gilb's categories also include processes and organizational aspects, as these have to be fulfilled as functions by the product and development. In addition to the requirement categories, Gilb describes the evolutionary nature of the design in conjunction with the requirements. According to Gilb, design in its evolutionary form poses requirements in and by itself as design decisions will inadvertently create restrictions and hence requirement-like conditions.

In 2007, Bapat et al. (2007) proposed an additional categorization for requirements. Herein, they describe four different types of requirements: requirements on design attributes, requirements on the existence of objects and relationships, and requirements on function. These four requirements are strongly related to the actual design process and derived from the CAD (Computer Aided Design background of the authors. The purpose of their categorization and classification was the design of a computational framework that supports the design synthesis. Therefore, Bapat et al. designed a computational representation of each of those requirement categories and, with the connection, enabled the assessment of the synthesis of different but related design elements. With such an approach, requirements can be utilized for purposes of analytical nature, which is important to the work at hand as well.

Lastly, Salado & Nilchiani (2014) defined a categorization for requirements based on the model of human needs by Max-Neef (1989). This classification resulted from the predominant focus on the design perspective. To detach the focus from the design perspective and the thoughts about what elements enable the fulfillment of a certain requirement, the authors propose an alternative approach that focuses on the system in its totality instead. Salado & Nilchiani claim that none of the existent classifications "achieve [...] a proper description of what the system is intended to do, and instead [...] end up consistently increasing the number of requirements without satisfying any new stakeholder need" (Salado & Nilchiani, 2014). Hence, the authors derive a requirement framework from the area where they saw this exact lack of focus: the stakeholder needs. The four existential types of human needs, as stated by Max-Neef (1989), and as an extension of Maslow's Hierarchy of Human Needs (1943) (Figure 3.5 on the next page) provided the basis of the requirement categories by Salado & Nilchiani. These four types of needs are: Doing, Having, Interacting, and Being. As a result, Salado & Nilchiani (2014) define and outline four requirement categories: Functional requirements (Doing), Performance requirements (Being), Resource Requirements (Having), and Interaction Requirements (Interacting).

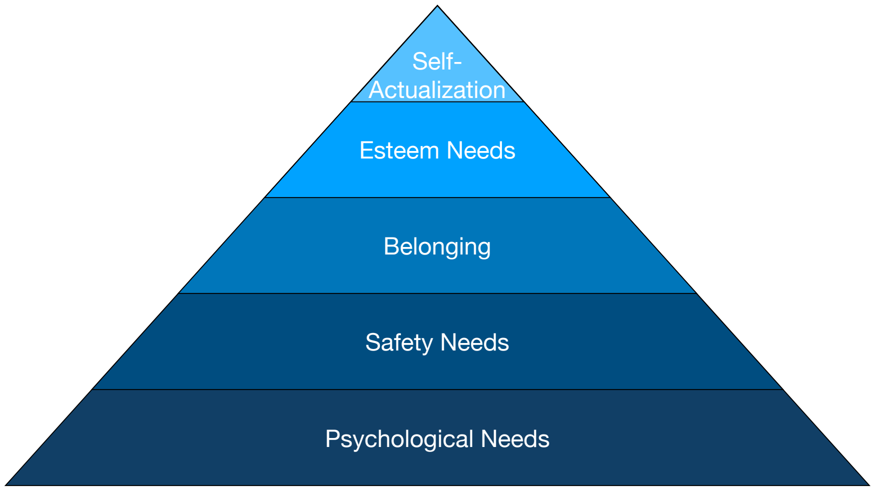

FIG. 3.5 - MASLOW'S HIERARCHY OF HUMAN NEEDS (MASLOW, 1943)

According to Salado & Nilchiani (2014), the four categories are enough to satisfy the elicitation and coverage of all needs of the shareholders while at the same time probing the add-on of requirements that are not necessary. In order to provide a hierarchical structure, Salado & Nilchiani (2014) propose three tiers of requirements: Break-Even, Goal, and Wish. The first tier herein is the minimum acceptable requirements that have to be fulfilled (e.g., functional ones that cannot be omitted). The second tier, the Goal Requirements, describes the desired value that the system should provide, and the third and last tier, the Wish Requirements, describes requirements that would be great to achieve but may be disproportionally difficult to fulfill. A similar structure can also be found proposed by Ehrlenspiel & Meerkamm (2013). With these two types of classifications, the approach allows for a shareholder and system-centric elicitation without directing the focus back to design elements. Furthermore, Salado & Nilchiani (2014) show a framework with sample requirements is functional and provides the claimed benefits.

Looking at the most recent publications and research regarding the topic of requirements and their structure, we can see trends that have also been partially observed for some of the aforementioned topics and sections (also see trends in Chapter 2). The first one is an increasing focus on Agile environments or development processes and the implications thereof when it comes to requirement classification in general and RE (Albuquerque et al., 2021; Amorndettawin & Senivongse, 2019; Aziz et al., 2017; Saher, Baharom, & Ghazali, 2017; Sunner & Bajaj, 2016). The key facts elicited from these publications are that there are various novel approaches that try and assess the compatibility of agile and classification of requirements. A driving factor was described by Amorndettawin & Senivongse (2019) as the authors describe that Agile inherits a tendency to focus on functional requirements due to its focus on iterative improvement and development of functions, whereas the non-functional requirements often are almost neglected. This circumstance can have grave implications and will be considered moving forwards as the compatibility with and applicability to agile environments and development projects were considered in the work of this dissertation.

The second trend is also shared with some of the topics from the other literature review parts: the application of neoteric and new computer and algorithm-supported methods to manage and classify requirements. This management includes all steps from the elicitation of requirements to the organization, prioritization, tracing, and change management (Hatıpoğlu, Atvar, Artan, Şereflışan, & Demir, 2017; Le, Le, David Jeong, Gilbert, & Chukharev-Hudilainen, 2018; Merugu & Chinnam, 2019; Qayyum & Qureshi, 2018; Singh, Singh, & Sharma, 2016; Win, Mohamed, & Sallim, 2020; Xiao, Zhou, & Sheng, 2016). Novel approaches try to automate the classification based on software and computer-based application such as text recognition. With these methods, the cited and referenced authors try to provide a general and objective analysis basis that can easily be applied due to its automatic nature. This idea and thought were also considered to enable universal applicability regarding the different philosophies behind agile and the "traditional" (Vierlboeck, Gövert, Trauer, & Lindemann, 2019) mentioned approaches, such as the Waterfall Model or Stage-Gate processes.

Besides the above-described trends, a few smaller accumulations have been found that all pertain to and address specific topics. They also, at least in part, align with other topics prior to this section. Yet, they are not as clear as the three described in the previous sections and will only be briefly mentioned here: various publications with a focus on security and safety were found (Alghamdi, Hamza, & Tamimi, 2019; Ali et al., 2018; Kamalrudin, Mustafa, & Sidek, 2018; Yahya et al., 2019) as and also some with sustainability and environmental aspects within requirements and their classification (Khalifeh, Farrell, Alrousan, Alwardat, & Faisal, 2020; Mireles, Moraga, García, & Piattini, 2017; Venters et al., 2017). These two trends show more general directions as the latter of the two, for example, attempt to incorporate sustainability and environmental consciousness into the domains and disciplines of RE. Therefore, not all approaches and publications are necessarily directly related to the requirements and their classification/categorization, but since they theoretically pose categories of their own and research is being conducted in all of these fields, and for all of these topics, they were chosen to be included here. These last two trends conclude the literature research of the most recent publications and scientific results.

All in all, as it has been outlined in the paragraphs above, the structure of requirements and distinctions are many-fold. Sometimes, as for the standards, the categories set boundaries while not being too restrictive to allow for a categorization that can be applied to as many projects and systems to develop as possible. In other cases, the categorizations are more general in nature and based not on universality but on different concepts instead. In order to define the core of all this, it is paramount to look at what the purpose of the research at hand is: the definition and assessment of requirement complexity when it comes to the development of systems. Therefore, the first discussed category, the standards, have to be taken into account for their universality yet restrictive inputs, while the other presented approaches have to be included for their generalist manner. Subsuming all the above-described research and publications, the following categorization is suggested for the research in this dissertation:

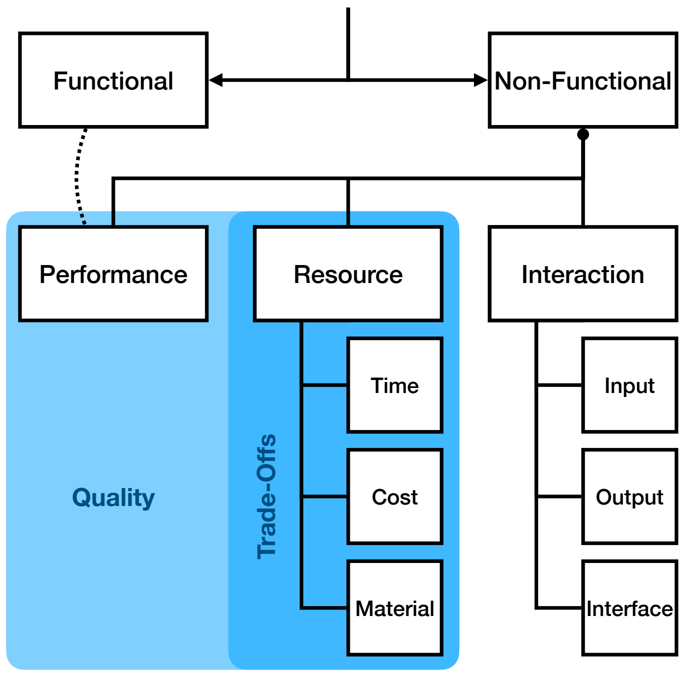

FIG. 3.6 - REQUIREMENT STRUCTURE DERIVED FOR THE RESEARCH

In the setup shown in Figure 3.6, two overall categories of requirements are used: functional and non-functional requirements. The former described all the essential functions and features that the system/product has to fulfill. The second type, the non-functional requirements, are supplemental or adjacent to the former. This means that, for example, a non-functional requirement can specify the magnitude of a functional requirement but can also provide additional and supplemental conditions. Therefore, non-functional requirements have to be further diversified. Three overarching categories were identified: performance requirements, resource requirements, and interface/interaction requirements. The first category is potentially directly related to the functional requirements because a performance requirement can but does not necessarily have to define the magnitude of the actual function. The second category, the resource requirements, are a diverse group, as they include physical as well as non-physical resource relations. Hence, they range from time resources to personnel and organizational resources as well as material, energy, and part requirements. This also means that resource requirements are strongly related to the above-mentioned trade-off type by Buede (2009). Together, the first two categories also define what was described above as quality since they define they directly influence the solution space and therefore user outcome. Lastly, interaction requirements define how the system/product is connected to adjacent components, elements, or environments, thus defining exactly how the exposure to the surroundings is to be interpreted. These categories of requirements are derived from all the above-stated and researched literature and serve as a comprehensive and holistic list. Since they incorporate the essence of all the above-analyzed sources, they contain thoughts of the standards, general industry practices, as well as the more generic systems approaches such as Buede's (2009) or Salado & Nilchiani's (2014). This way, it was possible to base the research at hand on a foundation that does not exclude any of the aforementioned sections while allowing for later application to all of them.

In conclusion, as described above, the work of Bapat et al. (2007) showed that with the right approach, requirements can be utilized to allow for computational or other theoretical analyses depending on their organization. This supports the core of the work at hand as it shows that an evaluation and interpretation of requirements constructs is possible and potentially fruitful as well as valuable. Hence, the above-described structure is utilized moving forward.

This concludes the research and state of the research regarding requirements, RE, structures thereof, and inherent complexity. The next chapter covers the last pillar of the literature review: Natural Language Processing (NLP) and its connections to other areas and topics.

BIBLIOGRAPHY OF CHAPTER 3

Abad, Z. S. H., Noaeen, M., & Ruhe, G. (2016, 12-16 September). *Requirements Engineering Visualization: A Systematic Literature Review.* Paper presented at the 2016 IEEE 24th International Requirements Engineering Conference (RE).

Albuquerque, D. W., Guimarães, E. T., Ramos, F. B. A., Costa, A. A. M., Gomes, A., Dantas, E., . . . Almeida, H. (2021). Disciplined or Agile?: Two Approaches for Handling Requirement Change Management. In M. Mora, J. M. Gómez, R. V. O'Connor, & A. Buchalcevová (Eds.), *Balancing Agile and Disciplined Engineering and Management Approaches for IT Services and Software Products* (pp. 130-150). Hershey, PA: IGI Global.

Alford, M. W. (1977). A Requirements Engineering Methodology for Real-Time Processing Requirements. *IEEE Transactions on Software Engineering, SE-3*(1), 60-69. doi:10.1109/TSE.1977.233838

Alford, M. W. (1978, 13-16 November). *Software Requirements Engineering Methodology (SREM) at the age of two.* Paper presented at the The IEEE Computer Society's Second International Computer Software and Applications Conference, 1978. COMPSAC '78.

Alford, M. W. (1985). SREM at the Age of Eight; The Distributed Computing Design System. *Computer, 18*(4), 36-46. doi:10.1109/MC.1985.1662863

Alghamdi, F., Hamza, N., & Tamimi, M. (2019, 1-3 May). *Factors that Influence the Adoption of Information Security on Requirement Phase for Custom-Made Software at SMEs.* Paper presented at the 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS).

Ali, I., Asif, M., Shahbaz, M., Khalid, A., Rehman, M., & Guergachi, A. (2018). Text Categorization Approach for Secure Design Pattern Selection Using Software Requirement Specification. *IEEE Access, 6*, 73928-73939. doi:10.1109/ACCESS.2018.2883077

AlZu'bi, S., Hawashin, B., EIBes, M., & Al-Ayyoub, M. (2018, 15-18 October). *A Novel Recommender System Based on Apriori Algorithm for Requirements Engineering.* Paper presented at the 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS).

Amorndettawin, M., & Senivongse, T. (2019, December). *Non-functional Requirement Patterns for Agile Software Development.* Paper presented at the 3rd International Conference on Software and e-Business, Tokyo, Japan.

Aziz, Y., Aziz, T., Malik, M. I., Baig, M. K., Ali, M. Z., & Baqer, M. (2017). Non Functional Requirement in Agile Software Development. *Technical Journal, University of Engineering and Technology (UET), Taxila, Pakistan, 22*(I-2017).

Bapat, V., Bettig, B., & Summers, J. (2007). *Requirements-driven design computations in next-generation CAD*. Paper presented at the 16th International Conference on Engineering Design - ICED'07, Paris, France.

Boehm, B. W. (1984). Verifying and Validating Software Requirements and Design Specifications. *IEEE Software, 1*(1), 75-88. doi:10.1109/MS.1984.233702

Buede, D. M. (2009). *The Engineering Design of Systems* (2nd ed.). Hoboken, NJ: John Wiley & Sons.

Bulusu, S. T., Laborde, R., Samer Wazan, A., Barrère, F., & Benzekri, A. (2017, 29 August - 01 September). *Which Security Requirements Engineering Methodology Should I Choose?: Towards a Requirements Engineering-based Evaluation Approach.* Paper presented at the ARES '17: Proceedings of the 12th International Conference on Availability, Reliability and Security, Reggio Calabria, Italy.

Bulusu, S. T., Laborde, R., Wazan, A. S., Barrère, F., & Benzekri, A. (2018). *A Requirements Engineering-Based Approach for Evaluating Security Requirements Engineering Methodologies*, Las Vegas, NV.

Callele, D., Wnuk, K., & Penzenstadler, B. (2017, 4-8 September). *New Frontiers for Requirements Engineering.* Paper presented at the 2017 IEEE 25th International Requirements Engineering Conference (RE).

Curcio, K., Navarro, T., Malucelli, A., & Reinehr, S. (2018). Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software, 139*, 32-50. doi:10.1016/j.jss.2018.01.036

Dalpiaz, F., & Brinkkemper, S. (2018, 20-24 August). *Agile Requirements Engineering with User Stories.* Paper presented at the 2018 IEEE 26th International Requirements Engineering Conference (RE), Banff, Alberta, Canada.

Dalpiaz, F., Ferrari, A., Franch, X., & Palomares, C. (2018). Natural Language Processing for Requirements Engineering: The Best Is Yet to Come. *IEEE Software, 35*(5), 115-119. doi:10.1109/MS.2018.3571242

Dalpiaz, F., Paja, E., & Giorgini, P. (2016). *Security Requirements Engineering: Designing Secure Socio-Technical Systems*. Cambridge, MA: MIT Press.

Darlington, M. J., & Culley, S. J. (2002). Current research in the engineering design requirement. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 216*(3), 375-388. doi:10.1243/0954405021520049

Department of Defense. (1969). Systems Engineering Management.

Dick, J., Hull, E., & Jackson, K. (2017). *Requirements Engineering* (4th ed.). Cham, Switzerland: Springer International Publishing.

Dresner, J., & Borchers, K. H. (1964). *Maintenance, Maintainability, and System Requirements Engineering*.

ECSS. (2009). Technical requirements specification. Noordwijk, Netherlands: European Space Agency.

Ehrlenspiel, K., & Meerkamm, h. (2013). *Integrierte Produktentwicklung* (5th ed.). München, Germany: Carl Hanser Verlag.

Elghariani, K., & Kama, N. (2016, 15-17 August). *Review on Agile requirements engineering challenges.* Paper presented at the 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia.

Franch, X., Ayala, C., López, L., Martínez-Fernández, S., Rodríguez, P., Gómez, C., . . . Rytivaara, V. (2017, 4-8 September). *Data-Driven Requirements Engineering in Agile Projects: The Q-Rapids Approach.* Paper presented at the 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), Lisbon, Portugal.

Garina, E. P., Romanovskaya, E. V., Andryashina, N. S., Kuznetsov, V. P., & Tsymbalov, S. D. (2021). Methodology for Creating a Product Development System Based on Complexity Management Strategy. In E. G. Popkova, V. N. Ostrovskaya, & A. V. Bogoviz (Eds.), *Socio-economic Systems: Paradigms for the Future* (pp. 1661-1669). Cham: Springer International Publishing.

Ghasemi, M. (2018, 21-21 August). *What Requirements Engineering can Learn from Process Mining.* Paper presented at the 2018 1st International Workshop on Learning from other Disciplines for Requirements Engineering (D4RE).

Gilb, T. (1997). Towards the engineering of requirements. *Requirements Engineering, 2*(3), 165-169. doi:10.1007/BF02802774

Gomes De Oliveira Neto, F., Horkoff, J., Knauss, E., Kasauli, R., & Liebel, G. (2017, 4-8 September). *Challenges of Aligning Requirements Engineering and System Testing in*

Large-Scale Agile: A Multiple Case Study. Paper presented at the 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW).

Hales, C., & Gooch, S. (2004). *Managing Engineering Design*. London, United Kingdom: Springer-Verlag.

Hatıpoğlu, P. U., Atvar, A., Artan, Y. O., Şereflışan, O., & Demir, A. (2017, 15-18 May). *Software requirement traceability analysis using text mining methods.* Paper presented at the 2017 25th Signal Processing and Communications Applications Conference (SIU).

Hsu, W., & Woon, I. M. Y. (1998). Current research in the conceptual design of mechanical products. *Computer-Aided Design, 30*(5), 377-389. doi:10.1016/S0010-4485(97)00101-2

IEEE. (1998a). IEEE Std 830 - IEEE Recommended Practice for Software Requirements Specifications. New York, NY: Institute of Electrical and Electronics Engineers.

IEEE. (1998b). IEEE Std 1233 - IEEE Guide for Developing System Requirements Specifications. New York, NY.

IEEE Standard Glossary of Software Engineering Terminology. (1983).

ISO/IEC/IEEE. (2011). ISO/IEC/IEEE International Standard 29148 - Systems and Software Engineering - Life Cycle Processes - Requirements Engineering: ISO/IEC/IEEE.

Jain, R., Chandrasekaran, A., Elias, G., & Cloutier, R. (2008). Exploring the Impact of Systems Architecture and Systems Requirements on Systems Integration Complexity. *IEEE Systems Journal, 2*(2), 209-223. doi:10.1109/JSYST.2008.924130

Kamalrudin, M., Mustafa, N., & Sidek, S. (2018). A Preliminary Study: Challenges in Capturing Security Requirements and Consistency Checking by Requirement Engineers. *Journal of Telecommunication, Electronic and Computer Engineering, 10*(1-7).

Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., & de Oliveira Neto, F. G. (2021). Requirements engineering challenges and practices in large-scale agile system development. *Journal of Systems and Software, 172*, 110851. doi:10.1016/j.jss.2020.110851

Kasauli, R., Liebel, G., Knauss, E., Gopakumar, S., & Kanagwa, B. (2017, 4-8 September). *Requirements Engineering Challenges in Large-Scale Agile System Development.* Paper presented at the 2017 IEEE 25th International Requirements Engineering Conference (RE).

Khalifeh, A., Farrell, P., Alrousan, M., Alwardat, S., & Faisal, M. (2020). Incorporating sustainability into software projects: a conceptual framework. *International Journal of Managing Projects in Business, 13*(6), 1339-1361. doi:10.1108/IJMPB-12-2019-0289

Khan, J. A., Liu, L., Wen, L., & Ali, R. (2019, 18-21 March). *Crowd Intelligence in Requirements Engineering: Current Status and Future Directions.* Paper presented at the 25th International Working Conference, REFSQ 2019, Essen, Germany.

Kolligs, J. W., & Thomas, L. D. (2020). The Origins of Requirements. *IEEE Systems Journal*, 1-11. doi:10.1109/JSYST.2020.2999557

Le, T., Le, C., David Jeong, H., Gilbert, S. B., & Chukharev-Hudilainen, E. (2018). *Requirement Text Detection from Contract Packages to Support Project Definition Determination.* Paper presented at the 35th CIB W78 2018 Conference: IT in Design, Construction, and Management, Chicago, IL.

Lindemann, U., Maurer, M., & Bran, T. (2009). *Structural Complexity Management*. Berlin Heidelberg, Germany: Springer-Verlag.

Loucopoulos, P., & Mylopoulos, J. (2003). Requirements Engineering editorial. *Requirements Engineering, 8*(1). doi:10.1007/s00766-003-0172-2

Maalej, W., Nayebi, M., & Ruhe, G. (2019, 25-31 May). *Data-Driven Requirements Engineering - An Update.* Paper presented at the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada.

Martins, L. E. G., & Gorschek, T. (2020). Requirements Engineering for Safety-Critical Systems: An Interview Study with Industry Practitioners. *IEEE Transactions on Software Engineering, 46*(4), 346-361. doi:10.1109/TSE.2018.2854716

Maslow, A. H. (1943). A Theory of Human Motivation. *Psychological Review, 50*(4), 370–396.

Max-Neef, M. A. (1989). *Human scale development: Concept application and further reflections*. New York, NY: Apex Press.

Mead, N. R. (2013, 15-19 July). *A history of the international requirements engineering conference (RE)RE@21.* Paper presented at the 2013 21st IEEE International Requirements Engineering Conference (RE).

Merugu, R. R. R., & Chinnam, S. R. (2019). Automated cloud service based quality requirement classification for software requirement specification. *Evolutionary Intelligence*. doi:10.1007/s12065-019-00241-6

Mireles, G. A. G., Moraga, M. Á., García, F., & Piattini, M. (2017, 21-24 June). *A classification approach of sustainability aware requirements methods.* Paper presented at the 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), Lisbon, Portugal.

Mufti, Y., Niazi, M., Alshayeb, M., & Mahmood, S. (2018). A Readiness Model for Security Requirements Engineering. *IEEE Access, 6*, 28611-28631. doi:10.1109/ACCESS.2018.2840322

National Aeronautics and Space Administration (NASA). (2020). *NASA Systems Engineering Handbook*. Washington, DC.

Niazi, M., Saeed, A. M., Alshayeb, M., Mahmood, S., & Zafar, S. (2020). A maturity model for secure requirements engineering. *Computers & Security, 95*. doi:10.1016/j.cose.2020.101852

Ponn, J., & Lindemann, U. (2011). *Konzeptentwicklung und Gestaltung technischer Produkte* (2nd ed.). Berlin Heidelberg, Germany: Springer-Verlag.

Qayyum, S., & Qureshi, A. (2018). *A Survey on Machine Learning Based Requirement Prioritization Techniques.* Paper presented at the 2018 International Conference on Computational Intelligence and Intelligent Systems, Phuket, Thailand.

Ramadan, N., & Megahed, S. (2016). Requirements Engineering in Scrum Framework. *International Journal of Computer Applications, 149*, 24-29. doi:10.5120/ijca2016911530

Saher, N., Baharom, F., & Ghazali, O. (2017, 25-27 November). *Requirement change taxonomy and categorization in agile software development.* Paper presented at the 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI).

Salado, A., & Nilchiani, R. (2014). A Categorization Model of Requirements Based on Max-Neef's Model of Human Needs. *Systems Engineering, 17*(3), 348-360. doi:10.1002/sys.21274

Schön, E.-M., Sedeño, J., Mejías, M., Thomaschewski, J., & Escalona, M. J. (2019). A Metamodel for Agile Requirements Engineering. *Journal of Computer and Communications, 7*(2), 1-22. doi:10.4236/jcc.2019.72001

Schön, E.-M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces, 49*, 79-91. doi:10.1016/j.csi.2016.08.011

Schön, E.-M., Winter, D., Escalona, M. J., & Thomaschewski, J. (2017, 22-26 May). *Key Challenges in Agile Requirements Engineering*, Cologne, Germany.

Singh, P., Singh, D., & Sharma, A. (2016, 21-24 September). *Rule-based system for automated classification of non-functional requirements from requirement specifications.* Paper presented at the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India.

Sunner, D., & Bajaj, H. S. (2016). Classification of Functional and Non-functional Requirements in Agile by Cluster Neuro-Genetic Approach. *International Journal of Software Engineering and its Applications, 10*(10), 129-138.

ur Rehman, S., & Gruhn, V. (2017). Security Requirements Engineering (SRE) Framework for Cyber-Physical Systems (CPS): SRE for CPS. In H. Fujita, A. Selamat, & S. Omatu (Eds.), *New Trends in Intelligent Software Methodologies, Tools and Techniques* (pp. 153-163). Amsterdam, Netherlands: IOS Press.

Venters, C. C., Seyff, N., Becker, C., Betz, S., Chitchyan, R., Duboc, L., . . . Penzenstadler, B. (2017, 20-28 May). *Characterising Sustainability Requirements: A New Species Red Herring or Just an Odd Fish?* Paper presented at the 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS).

Vierlboeck, M., Gövert, K., Trauer, J., & Lindemann, U. (2019). *Budgeting for Agile Product Development.* Paper presented at the Proceedings of the Design Society: International Conference on Engineering Design.

Vierlboeck, M., & Nilchiani, R. R. (2021, 13 Sept.-13 Oct. 2021). *Requirement Engineering in the Age of System and Product Complexity – A Literature Review.* Paper presented at the 2021 IEEE International Symposium on Systems Engineering (ISSE).

Villamizar, H., Kalinowski, M., Viana, M., & Fernández, D. M. (2018, 29-31 August). *A Systematic Mapping Study on Security in Agile Requirements Engineering.* Paper presented at the 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA).

Wagner, S., Fernández, D. M., Felderer, M., Vetrò, A., Kalinowski, M., Wieringa, R., . . . Winkler, D. (2019). Status Quo in Requirements Engineering: A Theory and a Global

Family of Surveys. *ACM Transactions on Software Engineering and Methodology, 28*(2). doi:10.1145/3306607

Wagner, S., Méndez Fernández, D., Kalinowski, M., & Felderer, M. (2018). Agile Requirements Engineering in Practice: Status Quo and Critical Problems. *CLEI Electronic Journal, 21*(1). doi:10.19153/cleiej.21.1.6

Weber, M., & Weisbrod, J. (2003). Requirements engineering in automotive development: experiences and challenges. *IEEE Software, 20*(1), 16-24. doi:10.1109/MS.2003.1159025

Weidmann, D., Kattner, N., Hollauer, C., Becerril, L., Chucholowski, N., & Lindemann, U. (2016, 4-7 December). *Methods collection to support requirements engineering with focus on structuring and consolidation of requirements.* Paper presented at the 2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM).

Win, T. Z., Mohamed, R., & Sallim, J. (2020). Requirement Prioritization Based on Non-Functional Requirement Classification Using Hierarchy AHP. *IOP Conference Series: Materials Science and Engineering, 769*. doi:10.1088/1757-899x/769/1/012060

Wymore, W. A. (1993). *Model-Based Systems Engineering* (1st ed.). Boca Raton, FL: CRC Press.

Xiao, Z., Zhou, Z., & Sheng, B. (2016). A New Classification Analysis of Customer Requirement Information Based on Quantitative Standardization for Product Configuration. *Mathematical Problems in Engineering, 2016*. doi:10.1155/2016/7274538

Yahya, S., Kamalrudin, M., Sidek, S., Jaimun, M., Yusof, J., Hua, A. K., & Gani, P. (2019, 19 September). *A Review Paper: Security Requirement Patterns for a Secure Software Development.* Paper presented at the 2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS).

Zamudio, L., Aguilar, J. A., Tripp, C., & Misra, S. (2017, 3-6 July). *A Requirements Engineering Techniques Review in Agile Software Development Methods*, Trieste, Italy.

CHAPTER 4: NATURAL LANGUAGE PROCESSING

"Language is a part of our organism and no less complicated than it."

Ludwig Wittgenstein

The topic and field of NLP are included in this dissertation due to the aforementioned fact that requirements are provided in text form. While the exact shape of a specification can vary, the textual nature of the records requires a machine-assisted approach when automatic or semi-automated processing is needed. Thus, a brief historic overview (4.1) is provided first in this chapter before taking a closer look at the field that brings together NLP with the requirements aspects and engineering thereof (4.2). Building on the overview, the third section in this chapter assesses the different available approaches to identify possible contenders and allow for the derivation of a research gap as well.

From a high-level point of view, despite different existing definitions (Liddy, 2001), NLP is an approach that allows computers to analyze and understand language and text to conduct linguistic analyses as well as speech and text manipulation (Beysolow, 2018; Chowdhury, 2003; Liddy, 2001). Therefore, the analysis option within the NLP field is the automatic and computer-assisted alternative to manual processing by a human, which is why it is applied and researched in this dissertation.

4.1 General Aspects and History *

Looking at NLP from a general perspective, three domains emerge as the main drivers: *Linguistics*, *Computer Science*, and *Psychology* (Liddy, 2001). The first field, Linguistics, is concerned with the structural and formal aspects of language; the second one, Computer Science, focuses on the processing and structuring of data; and the last one, Psychology, contributes insights into cognitive processes and psychological models of language. As a result, two directions exist in NLP: language processing and language generation. The language processing analyzes existing text/speech in order to create a representation, whereas language generation addresses the opposite direction, creating text from representation. The topic at hand is related to the former of the two.

Taking a closer look at the historical origins of NLP, besides the domain origins, shows that NLP goes all the way back to the 1940s. In these early times, Machine Translation (MT) was explored, which is the root of NLP (Chowdhury, 2003; Liddy, 2001). The first MT descriptions go back to Weaver's article about translation from the year 1949 (later published as a book section in 1955) (Weaver, 1955). In the article, Weaver's thoughts on the possibilities and potential obstacles regarding the translation of languages by machines are outlined. This was arguably influenced by the war circumstances of the subsequent years and insights gained during decryption and enemy message interception, which are concepts that are reflected in Weaver's description as well (Liddy, 2001). As a result, research into MT began based on stochastic and statistical approaches that attempted to tackle issues such as different translations of words, meanings, and ambiguities, to name a few. It was soon discovered that the task might be more difficult than anticipated. Similar concerns were already mentioned in Weaver's conversations with Professor Norbert Weiner in 1947 (Weaver, 1955).

* The content of this section has been extracted literally or with minor editorial modifications from M. Vierlboeck, D. Dunbar, and R. Nilchiani (2022), "Natural Language Processing to Extract Contextual Structure from Requirements," published by IEEE.

Following the efforts from the 1940s and early 1950s, Chomsky published the idea of generative grammar in 1957 (Chomsky, 1957) as part of his "Syntactic Structures." The concept describes grammar as a certain set of rules that result in the constellations and combinations of words forming sentences in a given language. Chomsky breaks from popular theories of the time (e.g., Shannon's communication theory (Shannon, 1948)) by saying that the structure of language cannot be addressed with pure statistical or empirical methods (Dahl, 2013; Manning & Schütze, 1999). In addition, Chomsky continued to work on aspects related to generative grammar all the way into the 1960s (Chomsky, 1965), and his work ended up defining what is now considered the rationalist approach in NLP that was prominent until the mid 1980s (Jurafsky & Martin, 2008; Manning & Schütze, 1999). Furthermore, the rationalist concept is part of what is considered universal grammar that evolved over time with humans (Jurafsky & Martin, 2008). Figure 4.1 shows an example: the structure of a sentence according to Chomsky's approach and constellation: A sentence is divided into a noun phrase (NP) and an additional verb phrase (VP). The latter also includes the object as a noun phrase.

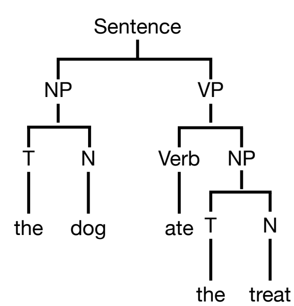

FIG. 4.1 - SENTENCE CONSTRUCT ACCORDING TO GENERATIVE GRAMMAR (CHOMSKY, 1957)

Throughout the 1960s, the movement based on Chomsky's approach of symbolic interpretation and the stochastic/statistical one based on Shannon's methods (Shannon, 1948) coexisted and advanced. Noteworthy results of this period were the first parsing systems by Harris (1962) as part of the symbolic paradigm. For the stochastic side, the first mention of

Artificial Intelligence (AI) emerged (Jurafsky & Martin, 2008), and Bledsoe and Browning (1959) developed the first optical character recognition approach based on the likelihood of each recognized string. Also in the 1960s, Woods published procedural semantics for a question-answering machine (Woods, 1968). Albeit still based on programmed subroutines, Woods' publications show elements that can be associated with Natural Language Understanding (NLU), as the answer to a question requires the extraction of semantic meaning from the question. The application was limited, but question-answer machines are still used today in voice assistants, for example.

During the 1970s and 1980s, the field of NLP grew broader, and topics such as NLU emerged, which considered aspects beyond NLP alone also in the direction of text/speech recognition and synthesis (Beysolow II, 2018; Rabiner & Juang, 1993). NLU was first approached and impressively demonstrated by Winograd (1972). In the publication "Understanding natural language," the authors demonstrate a program that is able to identify and select different shapes and colors in a simulated environment based on given text commands. This work bears strong ties with Woods' work mentioned above (Woods, 1968), and both drove the field of logic-based NLU. Additional noteworthy contributions to this trend include Schank and his colleague's work on language understanding programs (R. C. Schank & Riesbeck, 1981; Schank, 1972; Schank & Abelson, 1977).

In the second half of the 1980s and early 1990s, statistical approaches re-emerged (Lee & Reddy, 1988) as the primary focus of NLP/NLU, moving away from the symbolic ideas mainly shaped by Chomsky (Chomsky, 1965; Manning & Schütze, 1999). This popularity of stochastic methods in speech/language processing was significantly driven by IBM's Thomas J. Watson Research Center (Jurafsky & Martin, 2008). The re-emergence came with novel speech-recognition models that sought to bring NLU and speech analysis together (Hirshman, 1989). Eventually, before the beginning of the twenty-first millennium, the described changes and popularity made probabilistic models the predominant force in NLP, and the rapid increase in computing power, as well as the expansion of the internet, created a need for language-based information processing and extraction (Jurafsky & Martin, 2008). The combination of these circumstances led to a more unified but changed field of NLP/NLU and eventually gave way to the rise of Machine Learning in the first decades of the twenty-first century.

In the last 20 years, the interest in NLP has further increased rapidly in conjunction with the stark adoption of Machine Learning (ML) (Beysolow II, 2018). The pace that the subject and topics had picked up by the end of the 1990s was unprecedented (Jurafsky & Martin, 2008), especially since the developments in the decades before were described as incremental by experts (Futures Group Glastonbury CT, 1987). As a result, numerous banks and datasets were published in a few years (Marcus, Santorini, & Marcinkiewicz, 1993; Palmer, Gildea, & Kingsbury, 2005; Pustejovsky et al., 2003). These banks were collections and trees that contained text structures with underlying semantic information and details about syntactics. With the help of such banks and trees, further advances in parsing, tagging, reference resolution (Kibble, 2013), and information extraction were also enabled (Jurafsky & Martin, 2008). In addition to the published banks, the ML applications incorporated models such as the Bayesian Analysis (Gelman et al., 2013) and maximum entropy to train systems to process text in accordance with semantic, morphological, and or syntactic parameters (Kibble, 2013). Notable results were significant improvement in various directions, even some of the aforementioned, such as disambiguation, the answering of questions by a machine, and summarization (Kibble, 2013). Until the time of this writing (November 2022), NLP and NLU are active topics of research, and computer linguistics in total is described as an active field in AI research (Ghazizadeh & Zhu, 2021; Young, Hazarika, Poria, & Cambria, 2018).

In summary, NLP has gone through various changes over time. It began with Machine Translation and stochastic approaches, then transitioned to semantic and symbolic methods. A broader expansion accompanying the emergence of concepts of NLU and speech recognition enabled regained popularity of stochastic approaches before rapid changes in computer hardware and expansion of the web supercharged the progress of NLP, NLU, speech recognition, and MT, soon followed and accompanied by ML and AI. Also, potential future developments have been explored and considered, as shown in Figure 4.2, based on the predictions and analyses by Cambria and White (2014), who predict a stop of the reliance on word-based techniques to utilize semantics more broadly and effectively:

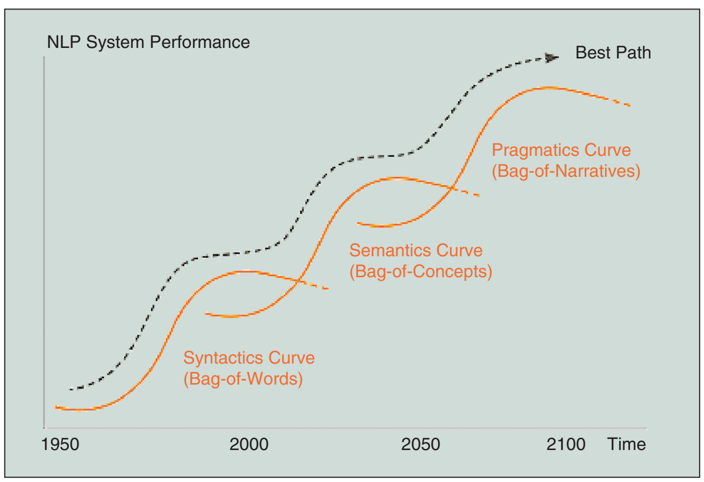

FIG. 4.2 - CONSIDERED EVOLUTION OF NLP OVER TIME (CAMBRIA & WHITE, 2014)

The history and progress outlined above help to understand the overall scope and changes that NLP has gone through over the course of its evolution. Since the topic of this dissertation was framed to address requirements engineering specifically, the two problems in conjunctions had to be researched as well. As shown in the next subsection, the application and research of NLP in the area of RE are not entirely new, and different approaches exist. Thus, an overview shall be given for this specific area as well.

4.2 Natural Language Processing for Requirement Engineering *

Looking at NLP in conjunction with RE, various approaches can be discovered as well, with some of them going back to the time that can be considered the mainstream beginning of RE in the late 1990s (Zhao et al., 2021). This long history and different directions have made the space of what is today called Natural Language Processing for Requirement Engineering - short NLP4RE - very diverse. As a result of this diversity, researching different approaches can be very difficult as not all solutions achieve popularity, mostly due to their niche existence and or special purpose. Thus, looking for approaches to consider and asses is a challenging task. Fortunately, studies have been conducted that target this issue. The most comprehensive study to date was published by Zhao et al. (2021). This study assessed the space of NLP4RE regarding tools, solutions, and techniques. The assessment included 404 relevant studies that the authors classified and used to extract 130 tools. These tools relied on 66 different approaches and 25 NLP resources (Zhao et al., 2021). In addition, Zhao et al. emphasize that most of the tools and techniques have not made it out of laboratory settings, are commonly focused on the analysis of requirements, and require specifications (Ferrari, Zhao, & Alhoshan, 2021), which is in line with the research shown in the next subchapter. Due to its peer-reviewed nature, extensiveness, and confirmation by other publications (Alzayed & Al-Hunaiyyan, 2021; Schrieber, Anders, Paech, & Schneider, 2021) it forms a big part of the sources of the presented work. Also, the results and insights presented by Zhao et al. have been confirmed by the research of other authors (Bruel et al., 2021; Montgomery, Fucci, Bouraffa, Scholz, & Maalej, 2022), whose work and discovered results turned out to be overlapping subsets of Zhao et al.'s. The final outcome and considered publications of Zhao et al. are depicted in Figure 4.3 on the next page.

* The content of this section has been extracted literally or with minor editorial modifications from M. Vierlboeck, D. Dunbar, and R. Nilchiani (2022), "Natural Language Processing to Extract Contextual Structure from Requirements," published by IEEE.

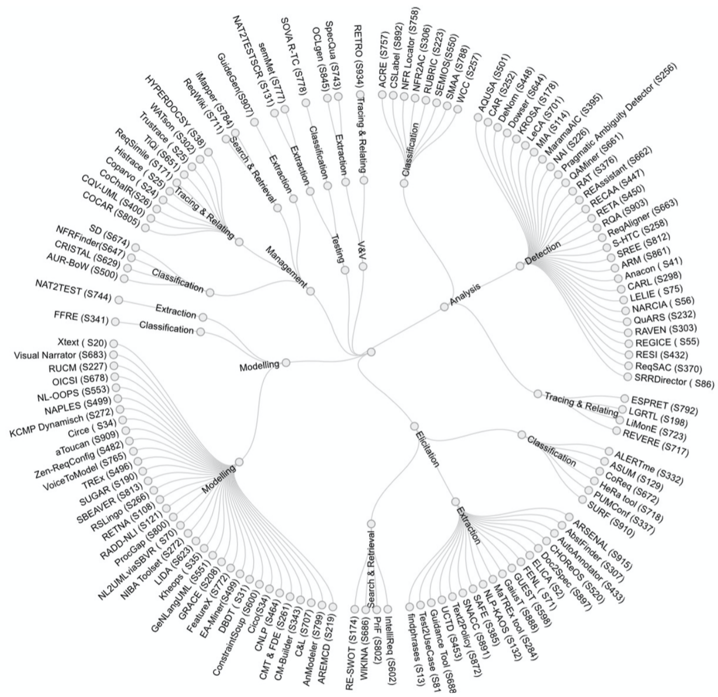

Fig. 4.3 - 130 NLP tools discovered and classified by Zhao et al. (Zhao et al., 2021)

In addition to the publication above, a separate round of tool research had to be added due to the fact that the presented research was conducted after the mentioned publication. This additional round of tool and approach research also allowed for the identification of currently (as of November 2022) ongoing trends in the field of NLP4RE. As such, the additional publications shall be outlined hereinafter. First, Mengyuan et al. presented an approach that utilizes NLP to extract domain models for control systems (Mengyuan et al., 2021). Their approach is based on Rupp's template for requirements and allows for the extraction as well as visualization of models. Second, two tools addressing causality in requirements and the detection thereof were discovered. These two tools, CiRA (Fischbach et al., 2021) and CATE (Jadallah, Fischbach, Frattini, & Vogelsang, 2021), address causal relationships within requirements. These relationships are assessed as to which requirements cause or depend on others. Third, Sonbol, Rebdawi, and Ghneim published their approach called "ReqVec," that allows for the deduction of semantic relationships as well as the classification of requirements (Sonbol, Rebdawi, & Ghneim, 2020). This approach, based on "Word2vec," showed high efficiency in the presented tests and is also considered in the next section. Fourth, Schlutter and Vogelsang published their approach to trace the connections between requirements, which they call Trace Link Recovery (Schlutter & Vogelsang, 2020). This approach uses an explicit content description of the requirements in the form of a semantic relations graph that allows for the tracing of connections within. Lastly, van Vliet et al. (van Vliet, Groen, Dalpiaz, & Brinkkemper, 2020) presented an approach for NLP based on crowdsourcing to solve shortcomings of what they consider a lack of accuracy and reliability of current approaches. The approach is not strictly related to RE but nevertheless addresses requirements in their general for and their management.

All in all, the last paragraph shows active and ongoing research in the field of NLP4RE in addition to the mentioned comprehensive study. Furthermore, the different directions show that there are still various topics and different ideas being pursued. This further supports the purpose and ideas of the presented work, as structure and additional organization is valuable. Also, such structure and organization can contribute to currently identified challenges, as outlined by Kaddari et al. (2021).

To specifically assess the publications and research that exist in relation to the work in this dissertation, a more detailed assessment similar to the ones for complexity had to be conducted. This is due to the fact that, while NLP is the applicable solution to the problem at hand, not all facets and tools are useful or practical. Thus, the existing literature was assessed with a specific set of criteria (outlined in the next subsection) that allows for a better and more nuanced evaluation.

4.3 Structural Assessment of Requirements Through & With NLP

As described in the introduction chapter, the aspect of systems whose connection to complexity is to be examined are the textual requirements. The discussed topic of this chapter though, NLP, addresses many more aspects than what is necessary for the purpose at hand. Thus, the scope of the assessment in this section was set to evaluate and compare specifically all approaches that allow for the elicitation of structure from textual requirements through the application of NPL as per the research goals and objectives described in Chapter 5 as well.

To conduct the described review and analysis in a valid and scientific way, a methodology that allows for tracing and reproducibility is crucial. While the often described as "gold standard" of reviews, the systematic literature review (Davis, Mengersen, Bennett, & Mazerolle, 2014)(also applied in Chapter 2) is a possibility, it is not usable in its full form due to the fact that the scientific area of NLP is diverse and shows a multitude of directions and options that not necessarily all are related to the purpose described in the introduction as part of the systems engineering domain. Including all branches and possibilities would create an almost insurmountable body of literature to assess. Thus, a semi-structured review, following the process of the development over time, would be the other systematic option. This possibility has been deemed not suitable for the task at hand though, despite the consideration of different concepts within the NLP space, due to the fact that the objective of the presented work is not the consideration of different methods but the comprehensive assessment of the space to gain insight into the overall situation. As a result, an integrative review has been chosen since it allows for the evaluation and conflation of existing literature regarding a matter and or popular topic to enable the derivation of different frameworks and viewpoints (Davis et al., 2014; Torraco, 2005). Furthermore, integrative reviews have been applied similarly in other contexts (Mazumdar, Raj, & Sinha, 2005; Torraco, 2005).

The integrative procedure of the literature review was conducted as follows: in the first step, the setting and frame were defined in addition to the selection of the sources to consider. With the sources at hand, criteria were defined according to which the analysis was supposed to be conducted. The set was derived from the objective, namely NLP application and tools for the elicitation of structure in requirements and systems engineering. The exact selection and definition of each criterion, including the reasoning, is outlined in the next subsection and was closely tied to the content of Chapter 5 to fit the objectives. The criteria enabled the analysis, which forms the core of the integrative analysis and allows for the definition of an overview of the space and scientific field. With the analysis and insights, the discussion produces a comprehensive overview of the space, including potential shortcomings that could be addressed in the future, which are also considered for the specific objectives of this dissertation.

As discussed above, the criteria to assess the literature in an integrative way are the foundation for the analysis. In order to derive the criteria, a current project of the authors was used to define what would be necessary and required for an NLP approach in requirements engineering to enable the extraction of structure.

As a side note, it shall be mentioned that due to the application focus and direct usage intent, software/tool availability and modifiability were also considered since it cannot be assumed that an existing solution is necessarily applicable per se to all other scenarios. Therefore, a modifiable and available (or at least acquirable) tool/approach is essential. Seven criteria were defined. Listed on the next page, each criterion includes the reason and evaluation foundation to describe its origin.

1 - Possibility to elicit structure of a requirement body/specification - This first criterion forms the core of the review presented. Approaches are judged based on the ability they provide to extract structure from a set of requirements or specifications thereof.

2 - Open-source, acquirable, or accessible via API - Tools that are not accessible or usable for acquisition do only serve as concepts and thus have to be considered of reduced value. Information can still be gained, but not directly applied, which is why a direct application is preferable if possible.

3 - No Necessity for full or partial supervision/validation - Involving or requiring humans for supervision or result verification is, while sometimes necessary, a step that can severely limit the application of an approach and thus has to be optional if it can be avoided. This is also due to the fact that supervision can come with additional limitations and or prerequisites, which should be avoided to not interfere with the limitation criterion.

4 - Proof of concept existing/shown - A proof of concept is essential to show the capabilities of the tool/approach as a methodology without validation has to be considered conceptual until validated. As such, tools and solutions with proof of concepts are preferred.

5 - No input limitations regarding format and or structure - Input limitations, while necessary at times, greatly reduce the applicability of approaches, and if the limitations do not align with the input, a solution has to be considered not suitable without adjustment.

6 - Modern architecture or active development/support - The application and actual implementation stand and fall with the compatibility of the architecture and or programming language. If a solution is built on an architecture that is not current anymore, it has to be considered outdated. The only way an older or uncommon architecture can thus be usable is if proper support and active development are available.

7 - Modifiability/possibility for settings adjustment - Adjustability of an approach can be crucial when it comes to application outside of the initially intended use case. As such, this possibility plays an important role also since adjustability can impact and potentially change the severity of other limitation criteria through circumvention, which is why it should also be considered for future adaptation and transfer.

With the criteria, different approaches could be assessed and compared. To allow for a unified evaluation, the possible answers in Table 4.1 were set to ensure reproducibility. Naturally, due to the variety of the publications, some criteria were not possible to be assessed since the publications did not include sufficient information needed for judgment. In these cases, 'UNKNOWN' was used as a placeholder. This placeholder does not indicate the complete absence of information but denotes that an assessment of the respective criterion would only be possible with further information.

Table 4.1 - Criteria Options and Answers

Criterion		Possible Answers/Assessments			
A	Possibility to elicit structure	Yes	No	In-Part	Other
B	Open-source or acquirable	Open-Source (OS)	Commercial (Comm)	No	Other
C	Supervision requirement	No	Yes	In-Part	Other
D	Proof of concept	Yes	No	Theory	Other
E	Input limitations	No	Yes	Other	
F	Modern architecture	Yes	No	In-Part	Other
G	Modifiability	Yes	No	In-Part	Other

With the criteria and the rubric outlined, the analysis was conducted. Herein, the criteria were applied to the set sources. In order to allow for the best comparison between the assessed sources, each publication was assessed individually and rated as per Table 4.1.

With the inclusion of the additions mentioned in the last section, the analysis covered 134 publications with 136 tools. For two publications, the presented approach included two tools. These tools were assessed as one approach, and upon further examination, an individual evaluation would not have made a difference in the results. With the insights and statistics gained from the analysis, the next subsection discusses the exact results of the research.

In order to provide the results in a structured way, the following content is divided into three parts: the first outlines the quantitative results of the evaluation, providing statistics and numerical results of the analysis; the second presents specific publications that showed certain features and or attributes worth mentioning and why they stand out; lastly, the third part illustrates interpretations of the former two in the form of insights into the statistics and content which also includes the discussion of the results overall.

4.3.1 Quantitative Results and Statistics

Summarizing the results of the research, Figure 4.4 shows the pie charts of the different criteria excluding the 'UNKNOWN' answers, as in those cases, the publication did not allow for a proper assessment of the respective criteria due to missing information. This result applied to 20 cases of Criterion B, two (2) cases of Criterion C, two (2) cases of Criterion D, three (3) cases of Criterion E, and four (4) cases of Criterion F. The 'Other' category was used for results that did not fit into any of the other answers but indicated some work in line with the criterion.

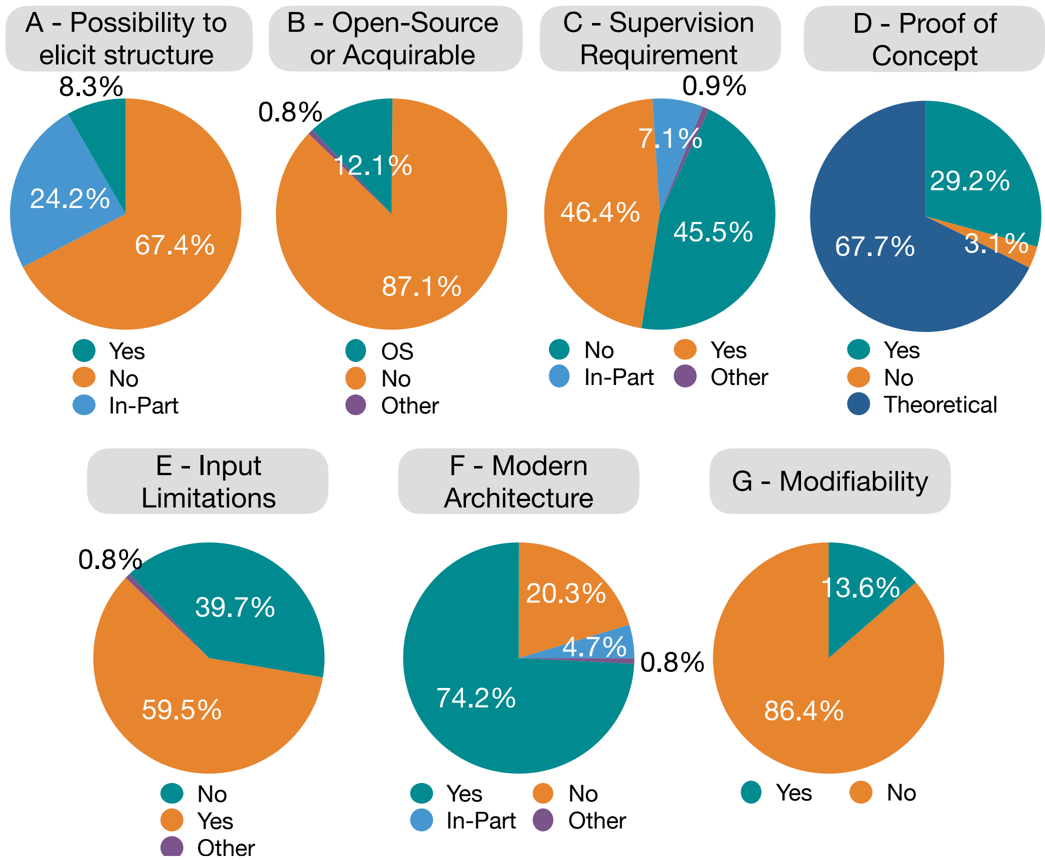

Fig. 4.4 - Numeric Evaluation Results

As Figure 4.4 depicts, the most important criteria (A and B) show a significant number of negative results. For example, over 67 percent of all assessed publications did not address the extraction of structure. Three publications were found that did not show any negative results (Ferrari & Gnesi, 2012; Sree-Kumar, Planas, & Clarisó, 2018; Tiwari, Ameta, & Banerjee, 2019), only partially positive ratings at most. In addition, ten publications were identified that had overall at least partially positive results except for Criteria B & G (Al-Safadi, 2009; Ferrari, Gnesi, & Tolomei, 2013; Hamza & Walker, 2015; Körner & Landhäußer, 2010; Li, Yue, Ali, & Zhang, 2019; Reinhartz-Berger & Kemelman, 2020; Sannier, Adedjouma, Sabetzadeh, & Briand, 2017; Schulze, Chimiak-Opoka, & Arlow, 2012; Sonbol et al., 2020; Tahvili et al., 2018), wherein G was in all cases a result of B due to the fact that no access to the source or algorithms does not allow for modification. The two groups are listed in Table 4.2 and 4.3.

Table 4.2 - Best rated publications

Publication		Criteria						
		A	B	C	D	E	F	G
Ferrari & Gnesi	2012	In-Part	OS	No	Theoretical	No	Yes	Yes
Sree-Kumar, Planas, & Clarisó	2018	In-Part	OS	No	Theoretical	No	Yes	Yes
Tiwari, Ameta, & Banerjee	2019	In-Part	OS	In-Part	Yes	No	Yes	Yes

Table 4.3 - Best rated not open-source publications

Publication		Criteria						
		A	B	C	D	E	F	G
Al-Safadi	2009	In-Part	No	No	Theoretical	No	In-Part	No
Körner & Landhäußer	2010	In-Part	No	No	Theoretical	No	In-Part	No
Schulze, Chimiak-Opoka, & Arlow	2012	In-Part	No	No	Theoretical	No	Yes	No
Ferrari, Gnesi, & Tolomei	2013	Yes	No	No	Theoretical	No	Yes	No
Hamza & Walker	2015	Yes	No	No	Theoretical	No	Yes	No
Sannier, et al.	2017	Yes	No	No	Theoretical	No	Yes	No
Tahvili et al.	2018	Yes	No	No	Yes	No	Yes	No
Li et al.	2019	In-Part	No	In-Part	Yes	No	Yes	No
Reinhartz-Berger & Kemelman	2020	In-Part	No	No	Theoretical	No	Yes	No
R. Sonbol G. Rebdawi N. Ghneim	2020	In-Part	No	No	Theoretical	No	Yes	No

The three publications in Table 4.2 show the highest potential, albeit none of them allows for full extraction of structure as per Criterion A. Nevertheless, these three contenders shall be assessed in more detail regarding their specific content below.

4.3.2 Individual Publication Discussions

The first publication from Table 4.2, published by Ferrari & Gnesi (2012), presents an NLP clustering algorithm titled "Sliding Head-Tail Component." This algorithm is supposed to analyze and understand requirement documents from a structural perspective to elicit cluster and relatedness information. In addition, the approach collates the structure of the document itself with the elicited one based on relations. The approach is also described to give recommendations or point to the find mismatches to improve the cohesiveness of the document. Inside the algorithm, the Sliding Head-Tail Component identifies requirements that are lexically related, which are clustered, and also sections that are lexically independent, thus allowing for partitioning (Ferrari & Gnesi, 2012). Through differentiation between related and unrelated requirements, the approach by Ferrari & Gnesi extracts what they call "hidden structure" (Ferrari & Gnesi, 2012), which is the structure resulting from the lexical information instead of the document. To achieve the lexical connections, Ferrari & Gnesi utilized two similarity metrics in conjunction: the first metric is based on the Jaccard index, and the second metric is based on the Levenshtein distance (Levenshtein, 1966). In testing, the approach showed promising results but was described to also have improvement potential due to the existence of potentially difficult false positives (Ferrari & Gnesi, 2012).

The second promising publication, published by Sree-Kumar et al. (2018), presented an approach to extracting a feature model from specification documents. The approach combines various methods and NLP tools to derive two algorithms as part of the open-source "FeatureX" (Sree-Kumar et al., 2018). The first algorithm extracts relationships within the document body and outputs heuristic results. These results are then used in the second algorithm to create a feature model candidate. The work by Sree-Kumar et al. showed promising results for the extraction of feature models and provided improvements over existing models.

Lastly, the most current contender is the 2019 publication by Tiwari et al. (2019). In their publication, Tiwari et al. outlined an approach that allows for the extraction of use case scenarios from requirements documents. While not the main focus of the approach, structure is extracted within the process as the input is handled by various NLP tools that feed a rule-based engine (based on heuristics (Tiwari & Gupta, 2015)). With the rule-based engine, use case names, actors, dependencies, basic flow steps, alternative flows, and post-conditions can be detected. The approach additionally includes the extraction of actual use cases which involves a human. The above-mentioned results of the engine are reminiscent of structure and could be used as such. Yet, the main purpose of the work presented by Taiwan et al. is use case extraction and analysis, of which the structure is merely seen as a byproduct.

4.3.3 Insights and Discussion

The above-presented results provide an overview of the space assessed by the literature review. Yet, the statistical results, while relevant, only show one perspective. Therefore, this subsection will address the insights and inferences based on the results.

To begin with, the results in the pie charts in Figure 4.4 show a significant number of negative results for all of the first three criteria. As such, it is apparent that most of the assessed approaches do not target the extraction of structure at all, and some only in part. Only 11 publications directly address the extraction of structure. This is in part due to the fact that RE, in general, has many different areas and topics that NLP4RE can address, such as classification or change management. Thus, structure extraction only forms one subtask that is affected negatively by the divided attention.

In addition, most publications were closed-source, and their basis was not available, which, for one, makes assessment and evaluation difficult and, two, completely prohibits use and modification without difficult and potentially erroneous reverse engineering. The fact that this applies to 115 of the assessed publications renders this group only useful for concept adaptation and potential transfer/recreation. This fact is also directly related and causes the high negative ratings for Criterion G as not accessible algorithms cannot be modified or checked regarding the possibility to make changes. Another relation of Criterion B exists with Criterion F, as older architectures that are not open-source are significantly more difficult to use. Fortunately, almost 80 percent of the assessed publications were built on modern architectures, at least in part.

Criterion B, the supervision requirement, together with Criterion E, the input limitations, shows another characteristic: a significant number of approaches (in the case of Criterion C, almost half) comes with limitations either on the input or the process side. This means that automation of the entire approach will be impossible before these limitations or required input can be solved or substituted. Yet, we argue that this is not a limitation for the extraction of structure.

Lastly, the criterion for proof of concept showed that there is a very low number of actual case studies that are being used for validation purposes. Since over one-third of all the assessed contenders showed only theoretical proofs of concept, the validity and real-world applicability have to be regarded as limited. Thus, even if an approach with a theoretical proof of concept were to be adapted or chosen, its correctness would require additional validation.

Adding to the statistical results, the predominately negative ratings of the most important criteria get exacerbated by the fact that, while the criteria individually might have shown some positive results, the overall and end result is even smaller. This can be seen in Tables 4.2 and 4.3, which include only 3 and 10 publications, respectively. This means that only 2.2

percent of all assessed publications were relevant at all based on the chosen criteria, and only 7.6 percent were remotely applicable. Therefore, less than 10 percent of the overall assessed literature showed relevance. To illustrate the effect that the overlap has, Figure 4.5 shows the sets of the most significant criteria.

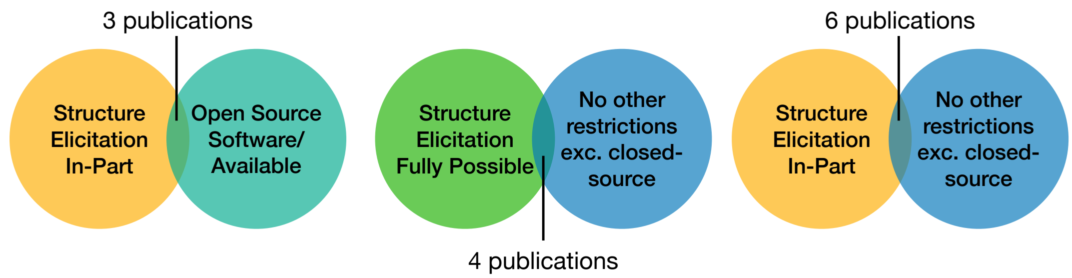

FIG. 4.5 - SETS AND OVERLAPS

As Figure 4.5 depicts, the overlap of the different sections is under 5 percent of the total number of publications for each combination. Thus, it can be deduced that not many publications at all pose potential solutions in a comprehensive manner. As a result, adaptations or additions are essential and cannot be avoided if a solution to the problem described at the beginning of this dissertation is to be attempted. Even the most promising contenders have limitations, as outlined hereinafter.

Despite three publications fulfilling all criteria at least in part, none of them targeted the extraction of requirement structure primarily: the first one mentioned, by Ferrari & Gnesi (2012), while allowing for the extraction of structure, mainly focused on the clustering and not the overall structure of the entire input in and by itself; the second contender, by Sree-Kumar et al. (2018), targeted the feature model as its primary objective and thus, structure was only a byproduct; lastly, Tiwari et al. (2019) addressed the extraction of use cases which included the elicitation of certain structural elements, but did not address structure comprehensively. Therefore, even the top contenders show drawbacks to be kept in mind.

This concludes the research and state of the research regarding Natural Language Processing and NLP4RE. As seen, the space of these two research areas can be considered fragmented at best, and various solutions exist for different problems. This variety was taken into account in the following chapters and is further discussed where applicable. Going back to the quote at the beginning of this chapter, we see clearly that the complications and differences within language are a characteristic that can create significant problems or inconsistencies, which is further discussed in the final chapters of this dissertation as well.

Dataset and Additional Sources

Note: The full dataset and source overview of the surveyed and evaluated literature/ publications (excluding the ones covered in 4.3, Table 4.2, and Table 4.3) can be found in the separate reference list in Appendix B.

Bibliography of Chapter 4

Al-Safadi, L. A. E. (2009). Natural Language Processing for Conceptual Modeling. *International Journal of Digital Content Technology and its Applications, 3*(3), 47-59. doi:10.4156/jdcta.vol3.issue3.6

Alzayed, A., & Al-Hunaiyyan, A. (2021). A Bird's Eye View of Natural Language Processing and Requirements Engineering. *International Journal of Advanced Computer Science and Applications, 12*(5), 81-90. doi:10.14569/IJACSA.2021.0120512

Beysolow II, T. (2018). What Is Natural Language Processing? In Applied Natural Language Processing with Python : Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing (pp. 1-12). Berkeley, CA: Apress.

Beysolow, T. (2018). What Is Natural Language Processing? In Applied Natural Language Processing with Python : Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing (pp. 1-12). Berkeley, CA: Apress.

Bledsoe, W. W., & Browning, I. (1959). *Pattern recognition and reading by machine*. Paper presented at the Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference, Boston, Massachusetts. https://doi.org/10.1145/1460299.1460326

Bruel, J.-M., Ebersold, S., Galinier, F., Mazzara, M., Naumchev, A., & Meyer, B. (2021). The role of formalism in system requirements. *ACM Computing Surveys (CSUR), 54*(5), 1-36.

Cambria, E., & White, B. (2014). Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]. *ieee Computational intelligenCe magazine, 9*(2), 48-57. doi:10.1109/MCI.2014.2307227

Chomsky, N. (1957). *Syntactic Structures*: Mouton & Co.

Chomsky, N. (1965). Aspects of the Theory of Syntax. Cambridge, MA: MIT Press.

Chowdhury, G. G. (2003). Natural language processing. *Annual Review of Information Science and Technology, 37*(1), 51-89. doi:https://doi.org/10.1002/aris.1440370103

Dahl, D. A. (2013). Natural Language Processing: Past, Present and Future. In A. Neustein & J. A. Markowitz (Eds.), *Mobile Speech and Advanced Natural Language Solutions* (pp. 49-73). New York, NY: Springer New York.

Davis, J., Mengersen, K., Bennett, S., & Mazerolle, L. (2014). Viewing systematic reviews and meta-analysis in social research through different lenses. *SpringerPlus, 3*(1), 511. doi:10.1186/2193-1801-3-511

Ferrari, A., & Gnesi, S. (2012, 24-28 Sept. 2012). *Using collective intelligence to detect pragmatic ambiguities.* Paper presented at the 2012 20th IEEE International Requirements Engineering Conference (RE).

Ferrari, A., Gnesi, S., & Tolomei, G. (2013). *Using Clustering to Improve the Structure of Natural Language Requirements Documents.* Paper presented at the International Working Conference on Requirements Engineering: Foundation for Software Quality.

Ferrari, A., Zhao, L., & Alhoshan, W. (2021, 25-28 May 2021). *NLP for Requirements Engineering: Tasks, Techniques, Tools, and Technologies.* Paper presented at the 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion).

Fischbach, J., Frattini, J., Spaans, A., Kummeth, M., Vogelsang, A., Mendez, D., & Unterkalmsteiner, M. (2021). *Automatic Detection of Causality in Requirement Artifacts: The CiRA Approach*, Cham.

Futures Group Glastonbury CT. (1987). State of the Art of Natural Language Processing (ADA188112). Retrieved from

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis* (3rd ed.). Boca Raton, FL: CRC Press.

Ghazizadeh, E., & Zhu, P. (2021, 2021//). *A Systematic Literature Review of Natural Language Processing: Current State, Challenges and Risks.* Paper presented at the Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1, Cham.

Hamza, M., & Walker, R. J. (2015, 17-17 May 2015). *Recommending Features and Feature Relationships from Requirements Documents for Software Product Lines.* Paper presented at the 2015 IEEE/ACM 4th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering.

Harris, Z. S. (1962). *String Analysis Of Sentence Structure*. The Hauge, Netherlands: Mouton Publishers.

Hirshman, L. (1989). *Overview of the DARPA Speech and Natural Language Workshop*. Paper presented at the Proceedings of the workshop on Speech and Natural Language, Philadelphia, PA.

Jadallah, N., Fischbach, J., Frattini, J., & Vogelsang, A. (2021). *CATE: CAusality Tree Extractor from Natural Language Requirements*. arXiv.org. arXiv.org.

Jurafsky, D., & Martin, J. H. (2008). *Speech and Language Processing* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Kaddari, Z., Mellah, Y., Berrich, J., Belkasmi, M. G., & Bouchentouf, T. (2021). *Natural Language Processing: Challenges and Future Directions*, Cham.

Kibble, R. (2013). *Introduction to natural language processing*. London, United Kingdom: University of London.

Körner, S. J., & Landhäußer, M. (2010). Semantic Enriching of Natural Language Texts with Automatic Thematic Role Annotation, Berlin, Heidelberg.

Lee, K.-F., & Reddy, R. (1988). Automatic Speech Recognition: The Development of the Sphinx Recognition System. New York, NY: Springer Science+Busienss Media.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady, 10*(8), 707-710.

Li, Y., Yue, T., Ali, S., & Zhang, L. (2019). Enabling automated requirements reuse and configuration. *Software & Systems Modeling, 18*(3), 2177-2211. doi:10.1007/s10270-017-0641-6

Liddy, E. D. (2001). Natural Language Processing. In *Encyclopedia of Library and Information Science* (2nd ed.): NY. Marcel Decker, Inc.

Manning, C., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. Cambridge, MA: MIT Press.

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank (MS- CIS-93-87). Retrieved from

Mazumdar, T., Raj, S. P., & Sinha, I. (2005). Reference Price Research: Review and Propositions. *Journal of Marketing, 69*(4), 84-102. doi:10.1509/jmkg.2005.69.4.84

Mengyuan, Y., Lisong, W., Jiexiang, K., Zhongjie, G., Wang, H., Yin, W., & Buzhan, C. (2021, 23-26 April 2021). *Automatic Generation Method of Airborne Display and Control System Requirement Domain Model Based on NLP.* Paper presented at the 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS).

Montgomery, L., Fucci, D., Bouraffa, A., Scholz, L., & Maalej, W. (2022). Empirical research on requirements quality: a systematic mapping study. *Requirements Engineering, 27*(2), 183-209. doi:10.1007/s00766-021-00367-z

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics, 31*(1), 71-106. doi:10.1162/0891201053630264

Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., . . . Lazo, M. (2003). *The TIMEBANK Corpus.* Paper presented at the Proceedings of the Corpus Linguistics Conference.

R. C. Schank, & Riesbeck, C. K. (1981). *Inside Computer Understanding*. New York, NY: Psychology Press.

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall.

Reinhartz-Berger, I., & Kemelman, M. (2020). Extracting core requirements for software product lines. *Requirements Engineering, 25*(1), 47-65. doi:10.1007/s00766-018-0307-0

Sannier, N., Adedjouma, M., Sabetzadeh, M., & Briand, L. (2017). An automated framework for detection and resolution of cross references in legal texts. *Requirements Engineering, 22*(2), 215-237. doi:10.1007/s00766-015-0241-3

Schank, R. C. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology, 3*(4), 552-631. doi:10.1016/0010-0285(72)90022-9

Schank, R. C., & Abelson, R. P. (1977). Scripts, plans, goals and understanding: An inquiry into human knowledge structures. Oxford, England: Lawrence Erlbaum.

Schlutter, A., & Vogelsang, A. (2020, 31 Aug.-4 Sept. 2020). *Trace Link Recovery using Semantic Relation Graphs and Spreading Activation.* Paper presented at the 2020 IEEE 28th International Requirements Engineering Conference (RE).

Schrieber, H., Anders, M., Paech, B., & Schneider, K. (2021). *A Vision of Understanding the Users' View on Software.* Paper presented at the Joint Proceedings of REFSQ-2021

Workshops, OpenRE, Posters and Tools Track, and Doctoral Symposium, Essen, Germany.

Schulze, G., Chimiak-Opoka, J., & Arlow, J. (2012). An Approach for Synchronizing UML Models and Narrative Text in Literate Modeling, Berlin, Heidelberg.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*(3), 379-423. doi:10.1002/j.1538-7305.1948.tb01338.x

Sonbol, R., Rebdawi, G., & Ghneim, N. (2020, 1-1 Sept. 2020). *Towards a Semantic Representation for Functional Software Requirements.* Paper presented at the 2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE).

Sree-Kumar, A., Planas, E., & Clarisó, R. (2018). *Extracting software product line feature models from natural language specifications*. Paper presented at the Proceedings of the 22nd International Systems and Software Product Line Conference - Volume 1, Gothenburg, Sweden. https://doi.org/10.1145/3233027.3233029

Tahvili, S., Ahlberg, M., Fornander, E., Afzal, W., Saadatmand, M., Bohlin, M., & Sarabi, M. (2018, 16-20 July 2018). *Functional Dependency Detection for Integration Test Cases.* Paper presented at the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C).

Tiwari, S., Ameta, D., & Banerjee, A. (2019). *An Approach to Identify Use Case Scenarios from Textual Requirements Specification*. Paper presented at the Proceedings of the 12th Innovations on Software Engineering Conference (formerly known as India Software Engineering Conference), Pune, India. https://doi.org/10.1145/3299771.3299774

Tiwari, S., & Gupta, A. (2015). A systematic literature review of use case specifications research. *Information and Software Technology, 67*, 128-158. doi:10.1016/j.infsof.2015.06.004

Torraco, R. J. (2005). Writing Integrative Literature Reviews: Guidelines and Examples. *Human Resource Development Review, 4*(3), 356-367. doi:10.1177/1534484305278283

van Vliet, M., Groen, E. C., Dalpiaz, F., & Brinkkemper, S. (2020). Identifying and Classifying User Requirements in Online Feedback via Crowdsourcing, Cham.

Weaver, W. (1955). Translation. In W. N. Locke & A. D. Booth (Eds.), *Machine Translation of Languages*. Cambridge, MA: MIT Press.

Winograd, T. (1972). Understanding natural language. *Cognitive Psychology, 3*(1), 1-191. doi:10.1016/0010-0285(72)90002-3

Woods, W. A. (1968). *Procedural semantics for a question-answering machine.* Paper presented at the Proceedings of the December 9-11, 1968, fall joint computer conference, part I, San Francisco, California.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *ieee Computational intelligenCe magazine, 13*(3), 55-75. doi:10.1109/MCI.2018.2840738

Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E.-V., & Batista-Navarro, R. T. (2021). Natural Language Processing for Requirements Engineering: A Systematic Mapping Study. *ACM Comput. Surv., 54*(3), Article 55. doi:10.1145/3444689

CHAPTER 5: RESEARCH METHODOLOGY & CONTENT

"Wenn Sie die Art und Weise ändern, wie Sie die Dinge betrachten, ändern sich die Dinge, die Sie betrachten."

(When you change the way you look at things, the things you look at change.)

Max Planck

This chapter outlines the methodology of the research as well as its content, including its uniqueness to distinguish it from other contributions and those assessed in Chapters 2 through 4. Also, the foundation for the contributions is built by establishing the hypotheses to address in combination with their anticipated outcome and validation. As such, this chapter is divided into the following sections: Research Methodology (5.1), Research Gaps and Opportunities (5.2), Research Hypotheses (5.3), Validation Plan (5.4), Anticipated Outcome and Results (5.5), Contributions (5.6), and Uniqueness (5.7).

5.1 Research Methodology

Since the goal of this dissertation is the creation of a new framework and approach, a structured research methodology had to be chosen to be applied. In order to also achieve a good foundation for future application of the results, which would also mitigate limitations complicating the transition from theory into practice, a methodology was selected that shares a common ground with both sides: the practical and theoretical one. This methodology is the "Design Research Methodology" (DRM) by Blessing & Chakrabarti (2009). Said research methodology allows for systematic development of knowledge, including validation. Due to the close relationship of the presented research to system design and its development, the rigor and structure provided by the DRM align with the purpose and contribution. Since the DRM targets the improvement of the design process in general (Chakrabarti, 2010), the development of the developed approach is considered one of these goals, and as such, the research was structured as shown in Figure 5.1 and described below.

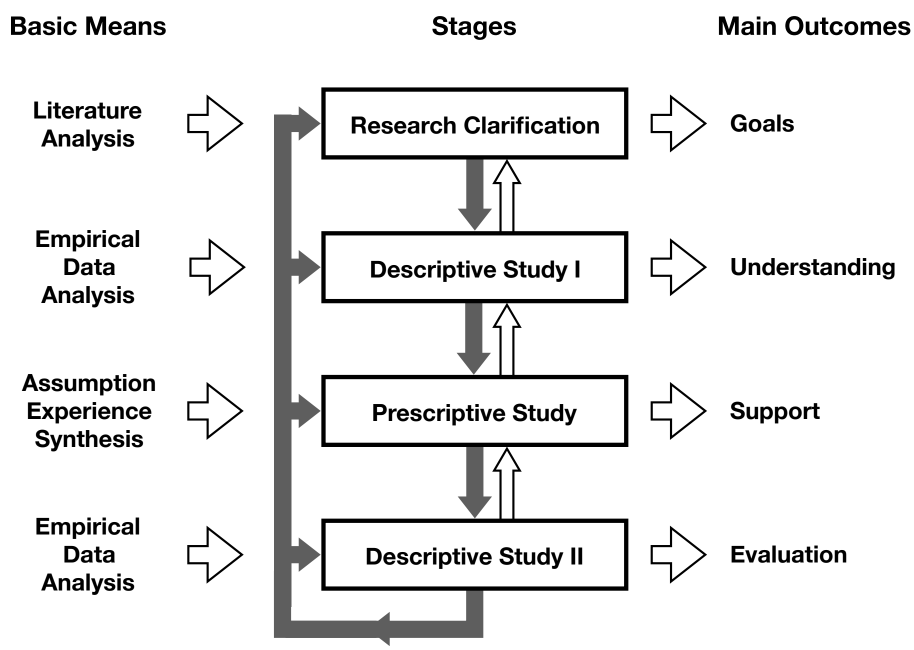

FIG. 5.1 - DESIGN RESEARCH METHODOLOGY FRAMEWORK (BLESSING & CHAKRABARTI, 2009)

Overall, the DRM is divided into four consecutive segments: Research Clarification, Descriptive Study I: Understanding, Prescriptive Study: Developing Support, and Descriptive Study II: Evaluating Support. In the first phase of the methodology, the Research Clarification, a literature analysis is used in order to determine which objectives pose a scientific value. These objectives then serve as the foundation to derive the respective goals and subgoals. In the following phase, the Descriptive Study I, a further and more thorough analysis, and research is conducted, including empirical data in order to determine the relevant influencing factors for the considered topic. This step is the basis of the research since it gathers all the information necessary for the consecutive analyses. Also, a profound understanding of the topic at hand is developed in this phase. In the phase of the Prescriptive Study, the gained knowledge is applied in order to synthesize the information foundation based on the given goals. This synthesis enables a specific solution generation as well as the derivation of the answers to the research questions/hypotheses. In the fourth and last phase, Descriptive Study II, the designed solution is evaluated and validated. Also, necessary improvements are derived where applicable (Blessing & Chakrabarti, 2009).

Based on the DRM, the research was structured to fit the methodology. Also, due to the direct relation of the methodology to real system and product design, the created approach and framework were modeled so that they allow for a practical application of the DRM within their boundaries since this would ensure a good basis for a transition to application.

The chapters and sections preceding this point lead to the end of the first Descriptive Study and thus the beginning of the Prescriptive Study. The literature analysis and empirical analysis have been conducted, as shown in Chapters 2 through 4. This, combined with the content of this chapter, now allows for the beginning of the actual solution synthesis and subsequent evaluation/validation. These steps and the content are described in further detail in the next sections, also in alignment with the applied methodology.

5.2 Research Gaps and Opportunities

With the outlined foundation and literature described in the previous chapters, the research gap to fill was defined as a starting point for the core work of the dissertation. Since the issues at hand converge the scientific fields of **complexity**, **RE**, and **NLP**, gaps for each are to be addressed. As such, the following paragraphs outline the gaps in these areas.

First, for the topic of **complexity** and also for system development, the literature reviews show that current approaches address complexity from a reactive standpoint or analogy, meaning that they either analyze already existing structures/architectures or draw from other similar projects/systems. This applies to all approaches except the Problem Complexity metric by Salado & Nilchiani (2014). While this last metric does not rely on an existing architecture, it does require the interpretation of conflicts within the design and thus cannot be applied without either interpretation or an architecture.

The reactive stance (from a development process point of view) and the reliance on existing information/interpretation create various issues regarding change management and locked-in costs. Hence, the gap existing here is an up-front approach that allows for the assessment of complexity based on factors that do not first have to be developed throughout the system design process. As such, requirements are to be utilized to design a complexity metric and framework that links the underlying complexity of requirements to the design and system complexity. This allows for prospective assessment and enables the RE contributions based on the gap below.

Second, for the field of RE, it has been shown that despite the long history and numerous approaches within the field, topics like complexity have been only side products and phenomena to be managed (Lindemann, Maurer, & Bran, 2009; Weidmann et al., 2016). The presented research adds an approach that allows for the inclusion of complexity within the RE process in order to transform complexity and related tipping points into parameters that can be evaluated and gauged instead of hedged and reactively handled. This allows for a more comprehensive RE process and reduction of risk due to emergence being factored into the process from the beginning onward.

The second gap is tightly tied to the first one regarding system and product complexity since the consideration, and potential management of complexity from the beginning, which is the requirement definition, onwards enables a more controlled and thus prospective approach since the need to develop and design certain features first is removed. Simply the assessment of complexity at the requirement stage can have huge benefits also regarding the aspects and circumstances mentioned in the introduction, such as cost and time.

For the last part, the NLP/NLP4RE field, Chapter 4 shows that given the criteria utilized, no available/accessible solution currently fulfills all of them sufficiently to provide a solution for the problem at hand to extract structure and contextual links from requirements. Hence, a lack of comprehensive applications that do not have major constraints and or limitations can be seen. Due to this gap, in addition to the attractiveness of the field and purpose, as also described in Chapter 4, an approach was developed to bridge the gap and allow for the extraction of structure and contextual links of requirement documents. Not only does such an approach enable the research in this dissertation, but also provides a better understanding of RE in the future based on information that is not implicit yet underpins requirement documents. Such an addition has significant potential benefits for system and product development in general, and it allows for risk reduction due to decreased uncertainty and lower chances of change necessities (also see Chapter 12 for further conclusions).

5.3 Research Hypotheses

Based on the gaps above, specific hypotheses for the research were defined to enable and guide all subsequent parts of this dissertation. As such, seven hypotheses were outlined as shown in Table 5.1, with their associated fields listed and colored respectively as well.

The hypotheses frame the purpose of this dissertation and cover the gaps listed in the last section. Furthermore, as indicated by the fields listed in the table, contributions in similar parts to each of the three areas underline the hypotheses. Hypotheses 6 and 7 form the core of the research and are the ones to be addressed last.

TABLE 5.1 - RESEARCH HYPOTHESES

#	Hypothesis	Field
1	Requirement text can be categorized and structured based on contextual and or explicit content.	NLP4RE
2	Structure and or networks can be derived from categorized requirement texts and content.	NLP4RE
3	A structure and frame for contextual interpretation and reasoning of requirements can be defined.	NLP4RE
4	The complexity of a requirement specification can be quantified based on the defined structure.	RE
5	The complexity of a requirement specification can be quantified in a general way.	Complexity
6	The complexity of requirements and the one of the system can be linked.	RE + Complexity
7	A higher level of requirement complexity increases the potential development effort/ costs.	RE + Complexity

Hypothesis 1 addresses specifically the extraction of structure from a given requirement body and specification. It sets forth the thought that it is possible to to categorize and structure requirements based on their implicit and or explicit information.

Hypothesis 2 builds on the first one and describes that the results of the categorization and structure elicitation can be used to build a network. Said network represents the underlying structure of the requirements either based on their implicit and or explicit information.

Hypothesis 3 adds the necessary information to the second one by specifying the necessary input that is needed to include contextual information in the interpretation and structure elicitation. The exact form and frame of this input is to be defined as well.

Hypothesis 4 builds on the results of the second one and states the quantification possibility for a structure that results from the analysis. With the input in the form of the structure, the different and suitable quantification possibilities shall be analyzed in this hypothesis.

Hypothesis 5 generalizes the content of the fourth one in order to remove potential limits that exist as a result of the reliance on structure and or topology. This hypothesis is closely related to the previous one and addressed in one effort. Yet, the two hypotheses serve different purposes and have individual contributions.

Hypothesis 6 poses the most ambitious and risky statement. Here, a connection between the requirements and the actual system and or its development shall be assessed and defined if possible. Due to the prescriptive and predictive nature of such a connection, empirical solutions were acceptable if discovered.

Lastly, Hypothesis 7 brings the effects and implications of numbers 4, 5, and 6 back to a process level (similar to how the fifth hypothesis builds on the fourth) to allow for a general interpretation and consideration. This hypothesis is also the base for practical application.

Figure 5.2 shows how all the hypotheses are connected.

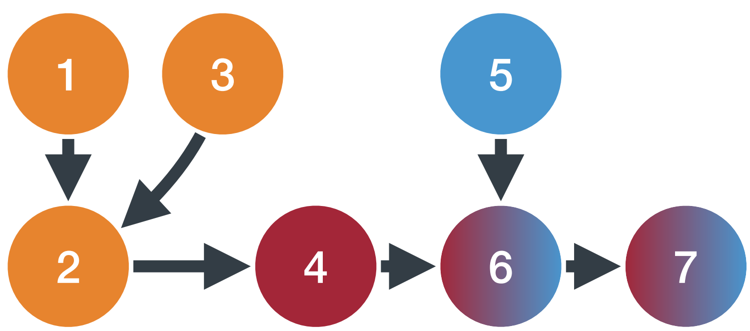

FIG. 5.2 - HYPOTHESES CONNECTIONS AND LINKS

As shown, Hypotheses 1 and 3 come together to enable Hypotheses 2, for which they can be considered prerequisites. The first three hypotheses combined feed into Hypotheses 4, which, together with 5, enables 6. Finally, based on the results and research conducted for Hypothesis 6, the seventh and last one is deduced.

5.4 Validation Plan

Since the validation of the hypotheses forms a critical part of every scientific contribution, the hypotheses in the last section were developed in tandem with their respective validation plans to be followed. These plans included the approaches as well as the necessary criteria. As such, this section outlines the validation approaches defined for each hypothesis that was later addressed with the results described in Chapter 11.

It must be noted that since the research in this dissertation concerns the entire development life cycle of a system, all steps, from the first definition of requirements all the way through production, implementation, and use, are factors to consider. As such, also considering the duration of such development projects, a validation with an ongoing project from the beginning on is not feasible given the time and doctoral program frame of this dissertation.

On a more general note, using actual systems and products for validation of research with similar scope to the presented one has been shown, for example, by Wertz & Larson (1996). The authors of the referenced book describe experiences stemming from space systems where they evaluated research findings and attempted validation by correlation and causality. Yet, the research in this dissertation does not utilize the analysis of existing data but instead defines a novel theoretic framework and foundation. Hence, such a validation is not suitable.

When we look for validation approaches of research similar to this dissertation, indirect approaches in academic settings can be found. Such approaches use case studies that are based on simulated notional systems or completed developments to apply the research results and assess the outcome of the metrics, either at the end or during the process. For instance, Dorsey et al. (2006) use a conceptual design in combination with a framework application to validate the latter by testing it with specific and pre-defined metrics. Roth & Mavris (2000) tried their developed approach by applying it to an existing system and comparing the calculated/predicted performance to the actual outcome. Brown & Eremenko (2008) used metrics to measure the hypothetical outcome of a design based on their research without actually going through the entire process or finishing the design. Furthermore, Saleh et al. (2003), Nilchiani (2005), as well as Long et al. (2007) used their developed approaches by applying them to existing systems in order to evaluate the outcome and juxtapose it with the known performance and metrics of the real systems. Lastly, Friedman & Sage (2004) stressed that case studies are vital for systems engineering research and as a result, outline a framework to design and apply case studies in this field as well as others.

As a result of the considerations and references above, the research in this dissertation uses a similar approach to the ones mentioned. This means that assumptions made during the solution generation are evaluated by comparing them to tested information, such as the data obtained by surveys, samples, or tests. The validity of the solution and created framework as such is tested through case studies. Due to the division into the different parts, the validation for each of the parts has been considered individually in addition to a complete validation for the entire framework. Table 5.2 shows the validation approaches for each part of the research, and additional details are provided below. Also, the hypotheses defined in 5.3 are indicated in the table to show which one is being addressed with which part of the validation.

TABLE 5.2 - RESEARCH HYPOTHESES AND VALIDATION APPROACHES

Framework Part	Hypo-theses	Assumption/Solution to validate	Validation Approach
NLP4RE and structure extraction	1	**Classification/characterization of requirement text content via NLP**	Surveys and logical reasoning through human checks and control of results
	2	**Extraction of structure/network from NLP approach and text corpus**	
	3	**Knowledge base structure definition**	
Requirement complexity metric	4	**Requirement structure complexity metric and quantification**	Case study and or mathematical reasoning/proof
System complexity and link to requirement complexity	5	**Requirement complexity quantification**	Virtual case study or validation through literature if chosen
	6 & 7	**Connections/Interactions between requirements and system development**	Virtual case study/studies (existing systems/products) and focus group(s)

For the first part, the NLP approach, three major assumptions and solution parts to validate can be seen: one is the classification and NLP categorization of the requirement text (Hypothesis 1), which is based on the rules set forth and implemented below as part of the knowledge base; the second part is the transition from the extracted structure to a network, which relies on the contextual connections (Hypothesis 2); third, the creation of a knowledge base concept and structure therein (Hypothesis 3). In order to validate these three parts, surveys and logical reasoning are applied partially with human control. This means that for both the categorization of the requirement text via NLP and the network extraction, sample results are to be cross-checked with the interpretation of one or more humans to validate the function of the approach. This will also allow for the discovery of potential unintended bias within the algorithm and software as well as extension possibilities for the knowledge base. For the latter, the validation follows a similar principle: a first knowledge base based on contextual information is validated by cross-checking the associations through human trials and evaluation, which in turn validates Hypothesis 3. For each of the three validation approaches above, blind evaluation is used, meaning that the tests are conducted without access to the control results of the framework parts to ensure as little bias as possible. In addition, for the third hypothesis, checks can be statistical as well as metric based, as shown by Derczynski and various others (Derczynski, 2016; Maynard, Li, & Peters, 2008; Robeer, Lucassen, Werf, Dalpiaz, & Brinkkemper, 2016). For these evaluations, recall, precision, F-Scores, and additional comparisons are used to assess the achieved results and validate them as per a threshold value.

For the fourth and fifth hypotheses, a case study is used in combination with mathematical approaches, where necessary. Such mathematical approaches and proof for a metric are conceptual validity checks for the developed topological complexity metric based on properties described as Weyuker's criteria (1988) (see Chapter 7). Furthermore, conceptual comparisons with other metrics existing in the literature are considered (Lindemann et al., 2009; Sinha & de Weck, 2013).

For the last two hypotheses, both targeting the interactions and connections between the system development and the requirement complexity, case studies are used as well, in combination with focus groups where necessary. The case studies are conducted to assess correlations, for instance, by analyzing the requirement complexity of samples and then conducting a correlation analysis as well as potential analyses regarding connections to other factors, where practical. In addition, another approach that was considered here is the assessment of causation, which was planned to be validated (only if possible) by the use of focus groups to trace and define actual connections that could also be quantifiably linked as a result (see Chapters 10, 11, and 12 for outcomes and implications).

As shown in the respective chapters, the last two hypotheses are subject to a plethora of limitations and, conversely bring many requirements for valid scientific assessments. As such, the focus groups could have served as a contributor to the validation where scientific and data-based validation might not be reasonable.

All in all, these validation plans were considered not just for the steps in which they are conducted but for the entire research process. Based on the content and plan above, the next section outlines the specific results that were anticipated for this dissertation.

5.5 Anticipated Outcome and Results

With the gaps and hypotheses, the anticipated outcome was defined to allow for the exact definition of the contribution. As already alluded to in the first chapter, the goal of this dissertation was to create a framework that enables the assessment of system complexity based on requirements from the early stages on. To make this possible, an approach was developed that analyses requirement text to elicit structure via NLP. Then, based on the structure, the complexity of the requirements was calculated, and its effect on the system development process was assessed. Eventually, gauging and managing potentially unforeseen behavior as well as acceptable levels of complexity of a system shall become possible.

To illustrate the concept's merits, Figure 5.3 is used to expand the specific contributions to the steps and the different subprocesses. In general, the conducted research adds a parallel process to the systems engineering development process (National Aeronautics and Space Administration (NASA), 2020), which introduces and creates complexity in the system. The parallel process has the three main aspects shown in the bottom stream of Figure 5.3 and contributes to the topics indicated by the numbers as explained below.

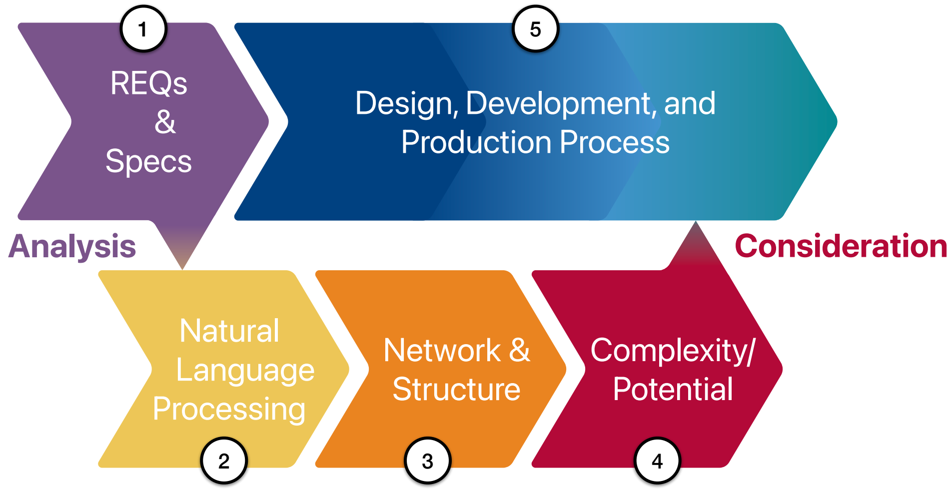

FIG. 5.3 - ANTICIPATED OUTCOME FRAMEWORK WITH CONTRIBUTION INDICATORS

(1) - For the area of requirements and RE, the conducted research targeted a complexity analysis metric that provides insights into the constellation and the underlying complexity of a set of requirements. This allows for an additional dimension of analysis and potential re-consideration of certain aspects, including visualization of the links as well as identification of the main drivers. This stands in close connection to the network and structure (Outcomes 3 & 4) as it is a direct result thereof. A possible representation of the outcome is shown below, which could also be added to trade space analysis as a third dimension, for example.

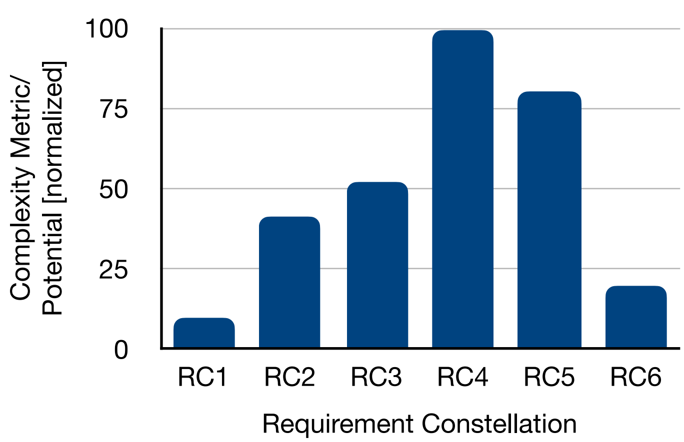

FIG. 5.4 - POSSIBLE REQUIREMENT COMPLEXITY OUTPUT

2 - Natural Language Processing, as shown in the literature review, despite existing in the RE space, has been characterized by solutions and applications that target specific applications. Due to the diversity of the complexity space shown in the literature review, in combination with the variety of requirements engineering standards and approaches, the limitations and risks of a targeted solution were considered and avoided where possible. As such, the anticipated outcome for the NLP aspect of the framework was a modularized approach that separates the knowledge base from the core algorithm, which provides the structure extraction from the requirement corpus. The knowledge base contains all reasons and information to classify and structure the input/corpus. In addition, by de-coupling the knowledge base from the algorithm, it can be modified, changed, and adapted where and when necessary. Figure 5.5 below illustrates this division and is further explained in Chapter 6.

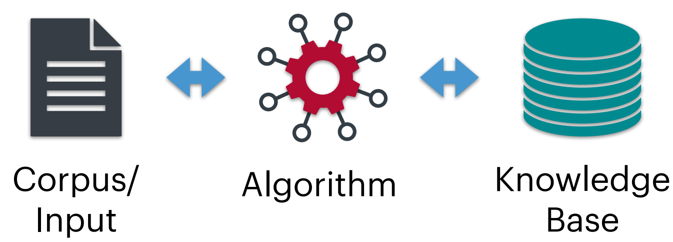

FIG. 5.5 - MODULAR NLP APPROACH CONCEPT

3 - For the network and structure part, the anticipated outcome was an interpretation of the NLP analysis that represents the connections underlying the requirements on a contextual level. As such, the analysis elicits information that is implicit and not necessarily in the text of the requirement corpus. For instance, the passenger capacity of an airplane and the engine power both have direct implications for the gross weight of the machine despite none of them necessarily indulging the word weight. The connection between those two values was planned to be elicited by the NLP and then transferred into a network. These networks can be multi-dimensional as well and even include specific themes, such as financial and budgeting aspects,

which could then be guided and adjusted via the knowledge base. Figure 5.6 below shows an anticipated representation connecting requirements to their underlying aspects (in this case, speed and weight of the system) in a network. Also, due to the source of the NLP, the type of connections in the network is available and could be used (further considered in Outcome 4).

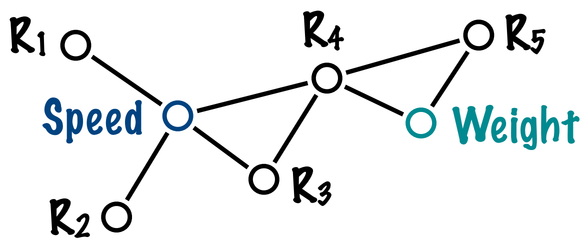

FIG. 5.6 - EXEMPLARY REPRESENTATION OF REQUIREMENTS AND CONTEXTS

(4) - For the complexity, the outcome anticipated was targeted towards the quantification aspect seen in parts of the literature review. As such, the outcome calculates the complexity potential contained and introduced by the assessed requirement constellation based on the network and structure. Various options for the actual quantification were considered, such as the entropy of the network, for example. The used metrics and formulas are dependent on Outcome 5 and were chosen accordingly.

(5) - Lastly, for the complexity that is to be gauged, the impact and implications are assessed as to how they can be considered in the development and design process. This inclusion was defined based on the effects that requirements have when it comes to the development process and is related to various aspects, such as change management, for example. In addition, a connection and potential correlation, where possible, was anticipated as the last core outcome. It is possible that the actual gauging of the complexity is treated flexibly in the future, but nevertheless, it was an anticipated outcome of this research.

Lastly, all of the above were integrated into an automatic framework that is not limited to one-time use. As such, it is possible to assess various constellations or even iterations throughout the development process. This iteration possibility creates compatibility with novel development models, such as agile models, since they fluidly change aspects of the design/requirements and are thus not necessarily bound to the standard succession of development steps. This compatibility is especially critical considering the trends related to Agile that were found and described in Chapters 2 and 3.

To summarize, the list below shows all the anticipated outcomes above succinctly:

I. Complexity analysis metric for RE and requirements

II. NLP approach to elicit structure from requirements/specifications

III. Network elicitation approach based on II.

IV. Potential complexity and metric thereof based on requirements and III.

V. Connection of VI. to the development and design process

VI. Automatization of the framework and holistic tool creation

The anticipated outcome and results, together with the hypotheses, formed the foundation for the conducted research and also guided the overall process. With the provided frame, all essential points could be addressed in a scientific and strategic way. Furthermore, the validation and verification of the results were enabled and both, the anticipated contribution as well as the hypotheses, are thus considered in the final judgment and measure of the completeness of the results.

5.6 Contribution

The contributions of this dissertation are twofold. For one, there are the main scientific advances of the work, which form the core and main results. On the other hand, application-related contributions also will most likely result from the planned work due to the close relationship between RE and applied methodologies. Therefore, this section outlines both of these contributions to describe the merits of the presented work.

As for the scientific contributions, the research in this dissertation provides a novel and prospective approach to address and gauge complexity all throughout but most importantly from the start of the system development process. As such, this research enables a more comprehensive assessment of complexity and provides a shift away from the reactive stance and architecture dependency, including all issues that come with it. Furthermore, an assessment from the beginning onwards also produces insights into the scientific parts of the process that will be researched. These parts are the extraction of structure from requirements based on their linguistic and contextual content and the calculation and quantification of the complexity within requirements. Lastly, the research in this dissertation contributes to the theoretical understanding and link between requirements, their complexity, and the system development and design process. These insights are enabled by the gauging and upfront assessment of complexity potential based on the requirements and quantification thereof. Overall, the research as a whole contributes to the mentioned fields and areas from the literature review chapters, which is visualized in Figure 5.7, also including the position of the seven hypotheses.

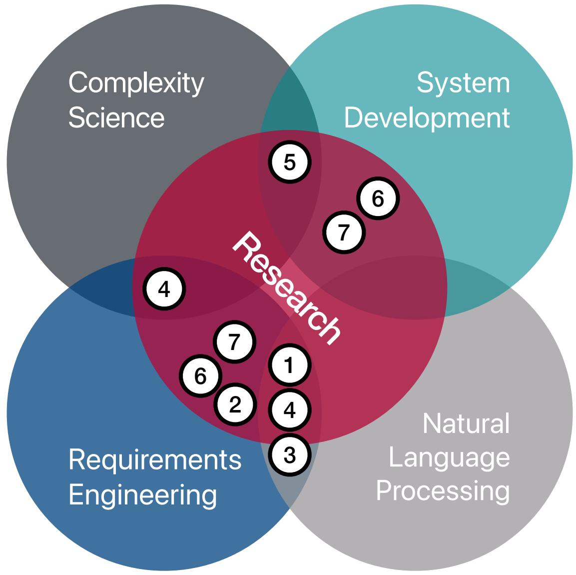

FIG. 5.7 - POSITION OF THE RESEARCH, INCLUDING HYPOTHESES

Regarding the contributions in a more application-related manner, the dissertation research provides potential and real-world problem-solving opportunities. First, the prospective and predictive gauging of complexity can provide directly applicable insights regarding ongoing system developments. Second, due to the parallel nature of the shown process and framework (Figure 5.3), the iterative application is also possible, which can be repeated throughout the development, and thus, the implications and effectiveness of changes can be assessed. Third, the separate building blocks of the framework also provide individually applicable benefits. On one hand, the NLP and network extraction provides an additional tool and potential visualization for the development and its current state based on the connections; on the other hand, the requirement complexity and quantification provides a novel benchmark and comparison parameter that can be used not just repeatedly or once within one development, but even across multiple projects to allow for comparison. Furthermore, other more minor aspects, such as the NLP knowledge base, are useful but require more specific applications and do thus not show as significant of a contribution.

Overall, this section shows that the twofold contribution of this dissertation does not necessarily limit it only to theory and scientific research but allows for (potentially even parallel) application, which is considered for the case studies and validation in Chapter 11.

With the contributions and core work outlined, the last section of this chapter addresses the uniqueness of the research and distinguishes it from existing work. This will also clearly outline the scientific advances of the dissertation in the fields analyzed in the literature reviews in Chapters 2 through 4.

5.7 Uniqueness of the Work

When looking at the scientific fields that surround this dissertation, combined with the assessments in the literature review, we see that the research does not open up an entirely new research field in and by itself. However, the content of the research is unique as it addresses unsolved problems and targets the generation of new insights and approaches as a scientific contribution overall.

The current research in the fields of RE and complexity focuses on analyzing the existing phenomena and dynamics, such as retroactive analysis and evaluation of complexity in system development. Hence, the current state of the art aims at creating insights and understanding that would allow for a future application based on analogy and similarity instead of addressing the root cause and allowing for extrapolation or predictions based thereon. The one publication closest related to this dissertation research, by Salado & Nilchinai (2014), shows a similar direction but, due to its limitations described in Chapter 2, does not go far enough to provide a comprehensive approach that would allow for an automatic assessment only based on requirements. While the efforts in the mentioned work are and were considered as far as the contained metric for problem complexity is concerned, it shows no significant overlap with the research in this dissertation.

The most distinguishing factor of the research in this dissertation is that it expands the capabilities of RE regarding the topics and aspects of complexity. This supports the reduction of risk, facilitation of better, simpler, and more robust design solutions, as well as overall development and design quality. As a result, this dissertation contributes to the complexity field as well as RE. Furthermore, due to the inclusion of the NLP approach outlined in Chapter 6, the presented research also shows distinct differences and contributions to the scientific field of NLP/NLP4RE. All in all, the most distinct attribute of this research lies in the novel combination of the outlined fields, which brings approaches previously not used in conjunction together to create new insights and scientific advances with practical application possibilities. Such a conjunction also potentially increases the chances of significance in scientific contributions, as evaluated and shown by Uzzi et al. (2013), depicted in Figure 5.8 below:

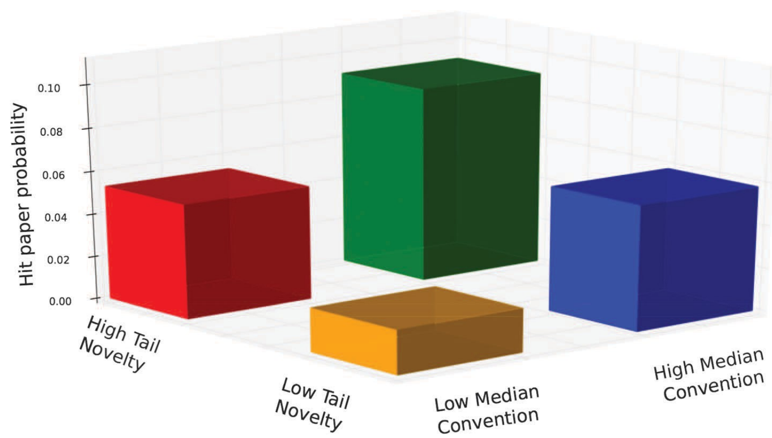

FIG. 5.8 - SIGNIFICANCE PROBABILITY BASED ON NOVELTY & CONVENTIONALITY (UZZI ET AL., 2013)

With the anticipated outcomes, contributions, as well as uniqueness described, the next chapter addresses the actual realization of the set goals and objectives that ensures a scientific and structured process that allows for replicability and traceability.

Bibliography of Chapter 5

Blessing, L. T. M., & Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. London, United Kingdom: Springer-Verlag.

Brown, O., & Eremenko, P. (2008). Application of Value-Centric Design to Space Architectures: The Case of Fractionated Spacecraft. In *AIAA SPACE 2008 Conference*.

Chakrabarti, A. (2010). A course for teaching design research methodology. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 24*(3), 317-334. doi:10.1017/S0890060410000223

Derczynski, L. (2016, May). *Complementarity, F-score, and NLP Evaluation*, Portorož, Slovenia.

Dorsey, J. T., Collins, T. J., Moe, R. V., & Doggett, W. R. (2006). Framework for Defining and Assessing Benefits of a Modular Assembly Design Approach for Exploration Systems. *AIP Conference Proceedings, 813*(1), 969-981. doi:10.1063/1.2169278

Friedman, G., & Sage, A. P. (2004). Case studies of systems engineering and management in systems acquisition. *INCOSE Journal of Systems Engineering, 7*(1). doi:10.1002/sys.10057

Lindemann, U., Maurer, M., & Bran, T. (2009). *Structural Complexity Management*. Berlin Heidelberg, Germany: Springer-Verlag.

Long, A. M., Richards, M. G., & Hastings, D. E. (2007). On-Orbit Servicing: A New Value Proposition for Satellite Design and Operation. *Journal of Spacecraft and Rockets, 44*(4), 964-976. doi:10.2514/1.27117

Maynard, D., Li, Y., & Peters, W. (2008). NLP Techniques for Term Extraction and Ontology Population. In P. Buitelaar & P. Cimiano (Eds.), *Ontology Learning and Population: Bridging the Gap Between Text and Knowledge* (pp. 107-127): IOS Press.

National Aeronautics and Space Administration (NASA). (2020). *NASA Systems Engineering Handbook*. Washington, DC.

Nilchiani, R. (2005). *Measuring Space Systems Flexibility: A Comprehensive Six-element Framework.* (Doctor of Philosophy). Massachusetts Institute of Technology,

Robeer, M., Lucassen, G., Werf, J. M. E. M. v. d., Dalpiaz, F., & Brinkkemper, S. (2016, 12-16 Sept. 2016). *Automated Extraction of Conceptual Models from User Stories via NLP.* Paper presented at the 2016 IEEE 24th International Requirements Engineering Conference (RE).

Roth, B., & Mavris, D. (2000). *Formulation of Available Energy-Based Methods to Assess System Affordability*. Paper presented at the NSF Grantees Conference, Vancouver, Canada.

Salado, A., & Nilchiani, R. (2014). The Concept of Problem Complexity. *Procedia Computer Science, 28*, 539-546. doi:10.1016/j.procs.2014.03.066

Saleh, J. H., Lamassoure, E. S., Hastings, D. E., & Newman, D. J. (2003). Flexibility and the Value of On-Orbit Servicing: New Customer-Centric Perspective. *Journal of Spacecraft and Rockets, 40*(2). doi:10.2514/2.3944

Sinha, K., & de Weck, O. (2013). *Structural Complexity Quantification for Engineered Complex Systems and Implications on System Architecture and Design.* Paper presented at the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2013, Portland, OR.

Uzzi, B., Mukherjee, S., Stringer, M., & Jones, B. (2013). Atypical Combinations and Scientific Impact. *Science, 342*(6157), 468-472. doi:10.1126/science.1240474

Weidmann, D., Kattner, N., Hollauer, C., Becerril, L., Chucholowski, N., & Lindemann, U. (2016, 4-7 December). *Methods collection to support requirements engineering with focus on structuring and consolidation of requirements.* Paper presented at the 2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM).

Wertz, J. R., & Larson, W. J. (1996). *Reducing Space Mission Cost* (1st ed.): Springer Netherlands.

Weyuker, E. J. (1988). Evaluating software complexity measures. *IEEE Transactions on Software Engineering, 14*(9), 1357-1365. doi:10.1109/32.6178

CHAPTER 6: STRUCTURE OF TEXTUAL REQUIREMENTS

"Model building is the art of selecting those aspects of a process that are relevant to the question being asked."

John Henry Holland

Based on the content and plan described in the previous chapter, this part of the dissertation is the first to describe solution generation and the development of novel results. To stick with the logical flow of the hypotheses, this chapter will address the structure within requirement specifications and how the elicitation thereof was achieved, including the knowledge base. Thus, this chapter is comprised of the following parts: NLP Foundation and Requirements Engineering Circumstances (6.1), NLP Tool Selection (6.2), NLP Approach Development (6.3), Network Approach Development (6.4), Knowledge Base Inclusion and Development (6.5), and Resulting Approach and Usage (6.6).

6.1 NLP Foundation and Requirements Engineering Circumstances

As the literature in Chapter 4 already showed, the space of Natural Language Processing is diverse, and a plethora of approaches can be found even for a task such as the definition of structure. As the review in Chapter 4 also showed, none of the approaches fulfill all the criteria and are thus applicable to the complete problem at hand, combining RE, complexity, and NLP. Thus, in order to systematically develop a fitting solution as per the problem-solving process by Ehrlenspiel & Meerkamm (2017), the problem and circumstances are to be defined first. Thus, this section addresses the general aspects behind text structure with consideration of the RE environment.

In general, when looking at NLP regarding structural analysis, the field applicable is text analysis. For this type of analysis, three categories exist (Achour, 1997): syntactic, semantic, and lexical methods. The first, syntactic approaches, are concerned with the structure of sentences and the grammatical constructs therein. The second, semantic approaches, address the logical structure of a sentence. While robust, semantic approaches require a set sentence structure, without which they cannot properly function. Lastly, lexical methods can be considered the most robust (Achour, 1997) due to the fact that they do not rely on part-of-speech analysis. Instead, lexical approaches work on the level of the character sequence to analyze the text.

To define the applicability of the approaches above and to see which is best suited for the task to elicit structure from a set of requirements, the limitations and restrictions that requirements bring have to be assessed. While, in theory, requirements, like every other body of text, are random in nature, they cannot or should not be structured as freely as continuous text passages, for instance. Looking at the standards mentioned in Chapter 3 or the NASA Systems Engineering Handbook (National Aeronautics and Space Administration (NASA), 2020), clear patterns emerge that requirements are recommended to follow. These patterns exist to prevent ambiguities, ensure completeness and verifiability, as well as introduce consistency while allowing for modifiability and traceability.

In addition to the patterns, some standards even specify the use of specific words that indicate different ranks or importance levels. The NASA Systems Engineering Handbook, IEEE Std. 830-1998 (IEEE, 1998), and the already mentioned ISO/IEC/IEEE 29148 (ISO/IEC/IEEE, 2011) recommend the verbs 'shall' for requirements, 'will' for fact statements or declarations of non-mandatory purposes, and 'should' for goals. Moreover, recommendations for numbered, short, single-sentence requirements can be found. These recommendations and patterns reduce the randomness of requirements significantly.

As a result of the specific composition of requirements, interpretation on a paragraph level is not necessary since the requirement statements are supposed to be independent and already semi-structured. Now, with the three types of analyses mentioned on the last page, it becomes clear that the following purpose order is applicable to the problem of structural elicitation from requirements: syntactic, lexical, and then semantic analysis. To explain this given the defined problem, examples shall be given for each.

Syntactic analysis forms the foundational layer of decomposition, meaning the structure and grammar of a sentence. With the increased rigidity of requirement statements, the syntactic analysis is simplified and thus can be conducted on a per-requirement basis. This allows for dependency definition, for example, in conjunction with part-of-speech tagging. For instance, a simple shall statement of a requirement can be analyzed syntactically regarding the parts within the sentence and how they grammatically relate to each other.

su can be added to syntactic analysis or conducted separately to deal with individual terms and their lexical environments. This can be used to connect synonyms or related terms, even if the exact word or form thereof is not identical. For example, two related terms that come from the same root word or family can be connected and detected through lexical analysis where possible and applicable.

Semantic analysis on the other hand, targeting the analysis of meaning behind the words and sentences, while not completely inapplicable to the tasks at hand, is the least important approach since the meaning becomes less important when sentences are disconnected and have to be valid and functional on their own. Thus, semantic analysis can be performed as a third layer in addition to the two previous ones to improve the quality of the results.

With the described reasons and applicability, the generation of an approach was conducted by working from the top down: first, a syntactic analysis was designed based on requirements, which can then be enhanced by lexical steps, and lastly, potential semantic evaluation. This layered structure and approach are shown in Figure 6.1 and form the basis as well as starting point for the entire NLP approach generation of this dissertation.

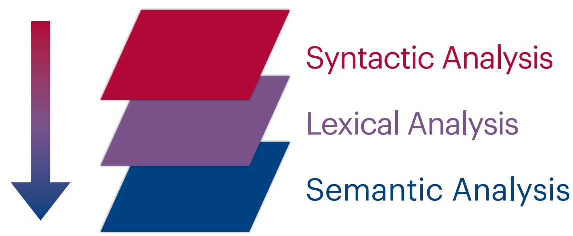

FIG. 6.1 - LAYERED STRUCTURE OF NLP ANALYSIS METHODS

Looking at specific requirement statements, the syntactic analysis to be conducted in a first step has to deconstruct the specific connections in a statement and or sentence. For instance, let's use the requirement "The Thrust Vector Controller (TVC) shall provide vehicle control about the pitch axis and the yaw axis" (National Aeronautics and Space Administration (NASA), 2020) as an example. This requirement consists of the TVC, the vehicle control, and the respective two axes. These four parts of the requirement form its structure and are called entities from here on. The syntactic analysis now needs to elicit the exact connections and potentially even the directionality of such links. For the given example, the connections are as follows: Thrust Vector Controller (TVC) is connected to vehicle control, which is connected to both the pitch axis and the yaw axis. This yields a chain with a fork at the end. Translated into a graph, the resulting structure is represented in Figure 6.2 below.

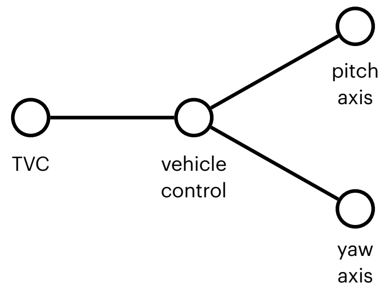

FIG. 6.2 - CONSTRUCT REPRESENTATION OF EXAMPLE TVC REQUIREMENT

Based on the example above, a general approach was defined that incorporates language specificity and grammar. In the example, the 'TVC' is the noun of the sentence, which is connected through the verbs 'shall' and 'provide' (herein, shall is an auxiliary of the main verb provide) to the compound direct object 'vehicle control,' which in turn is linked to the preposition objects 'pitch axis' and 'yaw axis.' Based on these sentence elements (e.g., noun, verb, object), which are called parts-of-speech (POS) in Natural Language Processing, the connections shown in the previous paragraph can be made on a general level, similar to what was shown in Figure 4.1. With these general connections, the network and graphical representation in Figure 6.2 expand into the generalized structure shown in Figure 6.3.

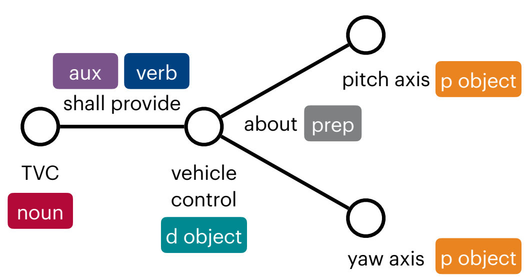

FIG. 6.3 - NETWORK REPRESENTATION OF EXAMPLE TVC REQUIREMENT IN GENERALIZED FORM

Given that requirements are more limited and organized than continuous text, in addition to the rules within the standards regarding clarity and prevention of ambiguities, the connections underlying the individual statements can be elicited based on the relationships of subjects/nouns, objects, and auxiliary factors. Thus, the approach designed was chosen to process statements individually and sequentially by connecting the subject to direct objects and prepositional objects in accordance with their respective preposition.

Yet, the described connections do not capture the following exceptions: subordinate clauses and lists. As the example shows, lists (either connected with the terms 'and' and or 'or') create forks in a graph representation; subordinate clauses, on the other hand, are connected to their respective main clause and can potentially break up the structure. To capture these cases, the rules in the following paragraphs were tested for their validity in the case studies.

For lists and enumerations, the connections cause the above-described forks or branches. This can be detected by the words 'and' or 'or' as well as the commas that divide lists and enumerations. One key aspect here is that not all commas indicate a list, but the two keywords above (together with other less common terms, such as 'as well as') indicate a preceding list if they contain what was above described as entities. Thus, the keywords, together with commas, can be used to detect lists and enumerations and elicit the respective connections. The exact implementation of this concept is described in detail in 6.3.

For subordinate clauses, detection can be addressed through a combination of secondary nouns and or objects. Two nouns can exist in a sentence and subsentence, for instance, with the requirement "The structure shall withstand loads of X, which can occur during landing." While not a particularly well-written requirement, the contained subordinate clause shows the secondary noun 'which' in conjunction with the verb 'can' and the prepositional object 'landing.' In this case, the structure is still affected by the term 'which,' referring to the load specified in the main clause. Thus, the statement structure is not sequential anymore. This can be implemented accordingly as well with the help of dependencies and was considered in section 6.3 with the later described limitations due to potential ambiguities.

Note that when it comes to language in general, randomness and anomalies that break the defined rules can occur due to the abundance of possible expressions. While this makes it difficult to claim full validity of any approach, a level of accuracy that is deemed acceptable was used. This also allows for the automatic application of the approach under consideration of the tested accuracy as otherwise, human intervention and or assistance would be necessary. While potential workarounds exist for these problems, such as the use of Machine Learning with a sufficiently large dataset, such possibilities would come with their own issues and could not claim 100 percent accuracy either. These factors and considerations were also kept in mind during the tool selection in the next section and are explained there in further detail as well.

With the described considerations, the design of the approach was conducted and tested in the case studies. Through parallel testing and subsequent refinement, the details of the implementation described in 6.3 were achieved with the tools in the next section.

6.2 NLP Tool Selection

With the foundation in the last section, the creation and development of the approach and framework was conducted. To address this issue, the right tools had to be researched and selected first. Then, with the tools at hand, implementation and testing were possible.

As already shown in the literature review chapter (Chapter 4), the existence of tools and options in the NLP space is not an issue. Yet, due to the fragmentation of the space and numerous niche solutions, an abundance of tools could possibly be used for the task to elicit structure from a set of requirements. Thus, in order to make an informed and replicable selection, a set of criteria was defined that guided and allowed for appropriate tool assessment.

First, in line with the evaluation in Chapter 4, the tools to use should be based on current (as of 2022) resources, such as coding languages and libraries. This reliance on contemporary tools not only ensures compatibility for the present and increases the chance of future support due to ongoing developments, but it also allows for the highest possible future upgradeability. Also, since funding was not available at the beginning of the development in late 2021, open-source and free solutions have been exclusively considered since other solutions, while potentially applicable, were not accessible. This exclusivity criteria also ruled out free yet proprietary solutions (e.g., with academic licenses) that could not be adapted or changed to fit the task and problem to be solved.

Second, due to the time restrictions of the dissertation and ongoing research that accompanied the approach and framework creation, tools with simple integration and little preparation requirements were prioritized. This was decided due to the fact that requirements already possess more structure than continuous text and thus, integration of features that might not be necessary for the long term would have created a potential waste of time and resources.

Third, tools with good documentation, support, and a high degree of community involvement were prioritized since the problem of eliciting structure from requirements had not been addressed before (see Chapter 4 and Appendix A for references), and thus, solving partial problems and relying on possible community support was a valuable asset.

Lastly, resource requirements and computing power were considered as secondary factors (not an exclusion criterion). Since the approach to develop was supposed to not be limited to an operating system or specific machine, the factor of manageable resource requirements was considered and taken into consideration. This also allowed for the possibility of running and collaborate in online environments and web applications.

With all these criteria, suitable options for software and libraries were researched, and the following contenders were identified:

- spaCy ("spaCy," 2022)
- Natural Language Tool Kit (NLTK) ("Natural Langue Toolkit," 2022)
- Stanford CoreNLP ("Core NLP," 2022)
- Apache OpenNLP ("OpenNLP," 2022)
- Transformer-based approaches, e.g., GPT-3 ("GPT-3," 2022)

Each of the listed tools was assessed regarding the criteria on the previous page. In the end, the choice fell on spaCy for the reasons described hereinafter.

First, spaCy was usable in Python, the coding language shared by other applied tools (see Chapter 7). Furthermore, spaCy contains a large pre-trained library, which does not require any pre-processing for over 59 languages ("spaCy GitHub page," 2022). The number of supported languages was an important factor since multi-language tools did not tie the developed approach to one language and also allowed for easier setup without the necessity to look for knowledge bases and libraries. The lack of manual preparation also meant avoiding potential licensing costs.

Second, spaCy performed very well as far as speed and accuracy are concerned in various benchmarks ("spaCy Facts & Figures," 2022). Speed was the deciding factor as it allowed for a swift setup, especially in combination with the spaCy's few implementation requirements and pre-processing necessary. In addition to the benchmarks, spaCy has also been shown to beat the other contenders when it comes to speed in studies (Choi, Tetreault, & Stent, 2015; Jugran, Kumar, Tyagi, & Anand, 2021; Vychegzhanin & Kotelnikov, 2019). The referenced studies were mainly focused on re-runs, and due to the nature of the work, such circumstances did occur, and thus, the advantage that spaCy provided was deemed important.

Lastly, spaCy's implementation, compared to a transformer approach, for instance, is substantially easier due to the possibilities of web applications, fewer resource requirements, and fewer dependencies to install/set up. Moreover, spaCy's instructions and tutorials are comprehensive, modifiable, and easy to adapt, which allows for an accelerated implementation.

While the choice of spaCy was suitable for the task at hand, future adaptations might be necessary and expansion possibilities were considered. For instance, while the application and use of spaCy were practical and helpful, transformer approaches, such as GPT, could enable more and additional features in the future. Furthermore, vector-based approaches pose another possibility for the knowledge base and data representation, for example. These opportunities are included in the last chapter of this dissertation.

With the spaCy library selected, the specific tools within the set that spaCy offers were evaluated. In a first step, to define the necessary accuracy, the different tiers that spaCy offers for pre-trained libraries were tested. This test with small requirement samples showed that the large core was necessary for spaCy to achieve an acceptable precision with no major identification errors. No significant differences could be found between the transformer core and the large core, which is likely due to the limited structure within requirements that does not benefit from a transformer-based approach compared to the large core.

As such, with the core selected and the spaCy setup completed, the algorithm and implementation were developed. This process and the results are described in the next section.

6.3 NLP Approach Development *

Now, with the selected tool and foundation, the development of the actual approach was conducted. Based on the structures and connections outlined in 6.1, the first step was to select the parts of the toolset that could be used in sequence or repetitively.

In accordance with spaCy's linguistic features, a small input text sample from the NASA Systems Engineering Handbook (National Aeronautics and Space Administration (NASA), 2020) was used to develop the process. The input was first split into its respective parts, meaning that a file was split into sentences that define the requirements. This also allows for the breaking up of entire files, such as tables, for instance. With this splitting, the individual requirements are tokenized, meaning that the sentences are divided into tokens, which are meaningful segments. These tokens include words, punctuations, and other elements. With these tokens, each individual element can be tagged or labeled based on its role in the context of the sentence, called part-of-speech (POS) tags. Herein, spaCy encodes strings as hashes to achieve low memory usage as well as improved efficiency. All of the outputs and results are stored in separate variables to allow for later access.

With the split requirements, tokens, and POS tags, the first step of structure generation can be conducted. To achieve this, the elements of the sentence that make up the nouns are combined with their description words. For instance, the term 'unmanned aerial vehicle' has as its root noun the 'vehicle.' Yet, using only the root as information would ignore all the adjacent descriptors, which is why a combination is necessary to retain all the information. As such, chunking is used to bring the different tokens together. The results of this step are the chunks in the individual requirements, which are possible contenders for entities in the structure to elicit and define.

* The content of this section has been extracted literally or with minor editorial modifications from M. Vierlboeck, D. Dunbar, and R. Nilchiani (2022), "Natural Language Processing to Extract Contextual Structure from Requirements," published by IEEE.

With the chunks as contenders, their role has to be identified. As already mentioned above and shown in Figures 6.2 and 6.3, the structure underlying the requirements connects subjects to objects and considers subordinate clauses as well. Thus, two aspects have to be considered: 1) the links between subjects and potentially multiple objects have to be identified, and 2) the detection and inclusion of subordinate clauses. This consideration was achieved by using Dependency Parsing, which enables the identification and labeling of dependencies within a sentence. The identified dependencies then enable the cross-connections and definition of which part of the sentence refers to and depends on another one. For example, the noun of a sentence connects to its verb, which in turn connects to the object. If there are prepositions, they act as a dependency relay. To visualize this, Figure 6.4 shows the first half of the dependencies of the TVC example requirement mentioned above.

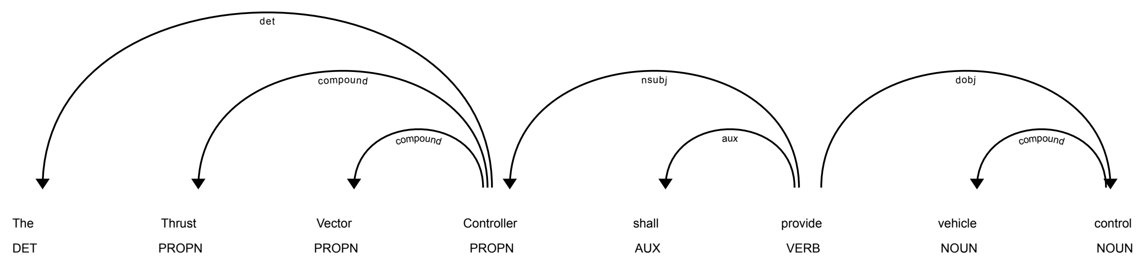

FIG. 6.4 - TVC REQUIREMENT DEPENDENCY STRUCTURE

As depicted, the controller serves as the main noun subject of the sentence and is connected to its compound additions ('Thrust' and 'Vector'), which also include the article. The verb 'provide,' which also has an auxiliary 'shall,' then connects the subject to the object 'control,' which again has an attached compound component 'vehicle.' With this structure, the logical flow of the sentence can be elicited, and the structures shown in Figures 6.2 and 6.3 can be elicited based on said logic.

The last two aspects to consider are the subordinate clauses and lists. To achieve the correct implementation and interpretation, rules were defined that identify a list or subordinate clause. Herein, the lists and enumerations were matched with the respective commas and words 'and' or 'or' and the subordinate clauses by an additional second or third noun that is subordinate to the main noun. By applying these rules, even sentences where the object appears before the noun (in passive cases, for example) can be identified.

Bringing all of these steps together in a logical flow allowed for the extraction of the necessary entities and chunks as well as logical order definition to elicit the structures shown in Figures 6.2 and 6.3. To visualize the overall flow, Figure 6.5 shows the individual sequential steps with their respective output. It has to be noted that while the flow includes a step called entity linking, this step is not to be confused with Named Entity Recognition (NER)(Mohit, 2014; Vychegzhanin & Kotelnikov, 2019), which is an approach to locate and identify important nouns as well as proper nouns and connecting them to a specific topic or context.

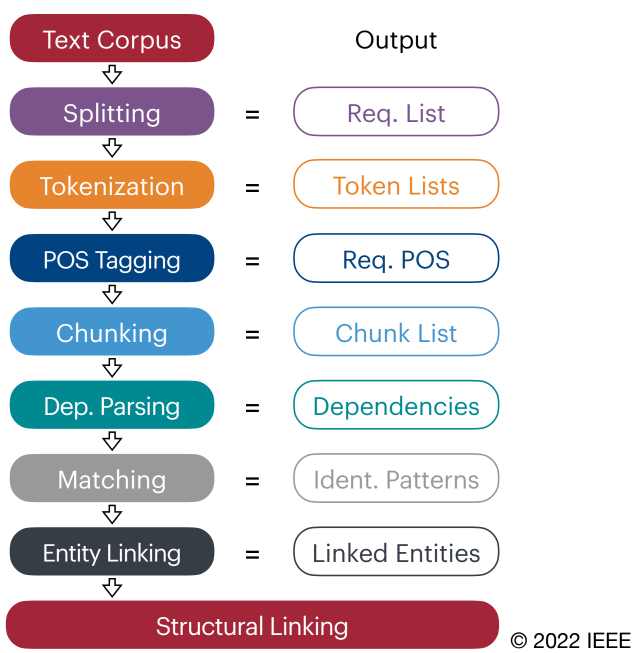

FIG. 6.5 - NLP APPROACH FLOW AND OUTPUT CHART (VIERLBOECK, DUNBAR, & NILCHIANI, 2022)

With the shown approach, the entities within the requirements can be elicited and linked according to the logical structure. The result of the code written in Python is displayed below for the TVC requirement as an example:

```
Requirement:
The Thrust Vector Controller shall provide vehicle control about the pitch axis and the yaw axis.
-----------------------------------------------------------------------------------------------
                   EntityText   TokenID    Dep   Entity#  Structure

0    Thrust Vector Controller         4  nsubj        1          1
1             vehicle control         8   dobj        2          2
2                  pitch axis        12   pobj        3          3
3                    yaw axis        16   conj        4          3
-----------------------------------------------------------------------------------------------
```

FIG. 6.6 - OUTPUT EXAMPLE OF DESIGNED NLP ALGORITHM

The output of the NLP algorithm shows all the elicited information as described in this section. Excluding the ID column, the first column lists the entity that was identified and its respective text. The second column notes the ID of the root token within the entity ('controller' is the fourth token in the sentence, for instance). The third column describes the found dependency type, as shown in Figure 6.4. The fourth column lists the entity number, and the last column defines the structure. In said structure, each number is connected to all subsequent numbers, i.e., 1 is connected to 2, which is connected to both 3s. Thus, the elicited structure is identical to the one depicted in Figures 6.2 and 6.3.

With the approach above, the foundation was provided that elicits the structure from each requirement individually. In order to combine all of the requirements in a subsequent step and form a compound network, an additional algorithm was required to bring together the output pieces. This approach, which forms the other half of the algorithm, is described in the next section before discussing the knowledge base.

6.4 Network Approach Development

With the information obtained from the NLP algorithm, the individual connections between the entities are defined. For instance, there are three connections shown in Figure 6.6: a connection between the Thrust Vector Controller and vehicle control, a connection between vehicle control and the pitch axis, as well as a connection between the vehicle control and the yaw axis. These connections can be produced as separate lines in a table, for example, and the algorithm was set up so that all connections from a requirement document are produced in one table without duplicates or circular connections where an entity connects to itself.

With the resulting table (called the 'from-to-list') that includes the information from which entity to which entity a connection exists, the network was produced that represents the entire structure of the requirement specification based on its entities. In order to achieve this network generation, the Python library NetworkX ("NetworkX," 2022) was used in conjunction with pyvis ("pyvis," 2022). These two tools were selected because the former was used previously in the third case study, which ensured compatibility without transfer, and the latter was chosen due to its native integration and seamless import of NetworkX network data.

With these two tools, the output of the NLP approach was used to generate a network file with the respective adjacency list and matrices as well as a web-based and interactive representation based on an HTML file. This way, not only could the data be best processed further (as described in the next chapter), but also be easily manipulated and potentially be used for different purposes. In addition to representation, NetworkX also allows for a multitude of calculations and analysis options, which is discussed in more detail in the next chapter.

Based on the from-to-list (called core list in the algorithm), networks and graph structures can be built and displayed. An example of such an output is shown in Figure 6.7 based on the requirements of the the Douglas DC-1 through DC-3 airplane family ("DC-1

Request for Proposal Gallery,”). These planes, introduced in 1935, have been continuously in use since then. Maybe surprisingly, the requirement specification for these airplanes was written on a single page and consisted of less than 150 words. This makes for an illustrative yet small enough example to illustrate the concepts. The only modification to the original text of the requirements was that they were transformed into individual statements. This was necessary since some of the original requirements were not written as statements, but as bullet lists, which makes them not suitable for the developed NLP approach and not compliant with modern RE standards. After the modification, the NLP process produced a result that was validated by human interpretation of the input.

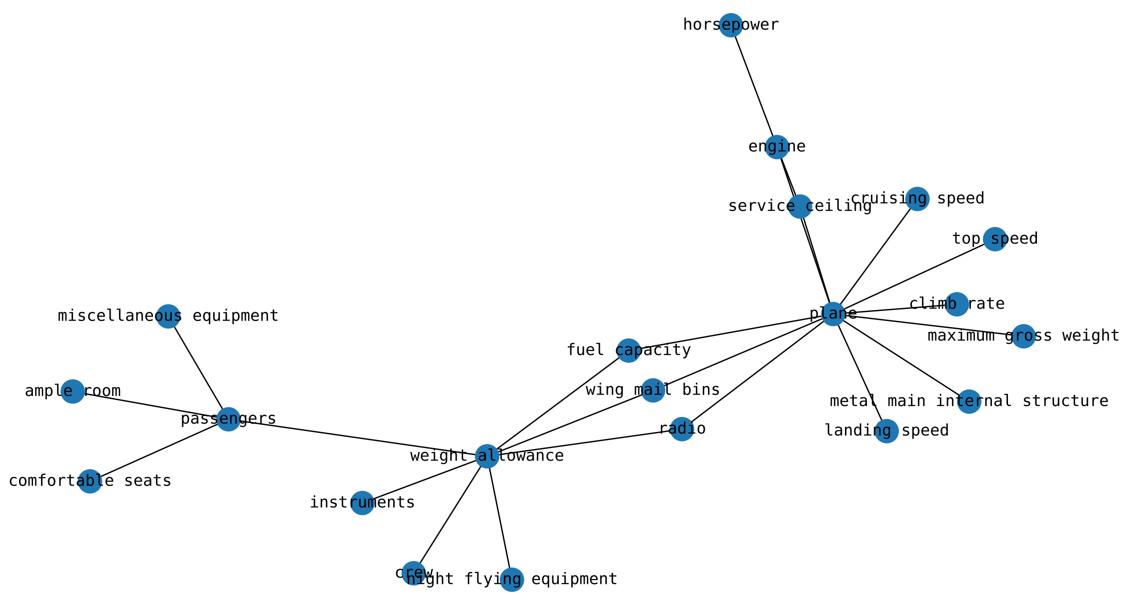

FIG. 6.7 - NETWORK OF DC PLANE FAMILY REQUIREMENT STRUCTURE

As seen in Figure 6.7, the output of the algorithm depicts all the entities and their respective links. This output can also be represented as an adjacency matrix or adjacency list that is used in the coming chapters for the complexity and quantification approach.

With the elicitation of the connections, not only can the network of the entities be defined, but with the additional information, which requirement contains which entities, the connections between the requirements can also be identified. To achieve this, the terms are assigned to the respective parent requirement, and based on the entity connections, the requirement connections can be built. Furthermore, if the source document contains a structure in the form of a hierarchy, i.e., numbers and identifiers of certain requirements, said structure could also be identified and considered.

The three described layers of structures and networks, as well as their respective information, are the result of the NLP approach. The layers can be used, elicited, and analyzed individually with only one dependence between the bottom and middle layers. Hence, the NLP approach that was created as part of the framework enables analyses on various levels that are addressed in the case studies as well. To visualize the layers and their connections in a flow chart, Figure 6.8 depicts the setup and resulting dimensions with their sources.

FIG. 6.8 - STRUCTURAL LAYERS AND DEPENDENCIES OF THE NLP RESULTS

With the NLP algorithm outlined and created, the second pillar shown in Figure 5.5 was added: the knowledge base and its implications. This aspect and the specifics are outlined in the next section.

6.5 Knowledge Base Inclusion and Development *

As already alluded to in Chapter 5, the knowledge base of the approach contains all the information necessary to conduct the structural elicitation. As such, the foundation and core of spaCy, for example, which is used within the NLP approach, is part of the knowledge base and assumed to be given, depending on the tool chosen.

Yet, another aspect of the knowledge base was considered and developed. This specifically pertains to the third contribution described in Section 5.5 and addresses the inclusion of context. This inclusion is not possible simply based on the core of the NLP tool, for example, since it requires additional analysis and cannot only depend on the identifications made by the NLP algorithm, for instance. Therefore, this part concerns the bottom half of Figure 6.1, the lexical and, in part semantic analysis of the content.

The inclusion of context in such networks was considered to achieve an extension and potentially also deduce lateral links that are implicit in nature. For example, different requirements could describe the mass of subsystems, and, as a result, all affect and influence the total mass of the system. However, these connections are not necessarily part of the requirements since the lower level ones could refer to 'payload' and 'capsule weight,' just to name a few, which would not have any connection as per the NLP. Hence, implicit connections can only be found and defined by adding additional information that allows for reasoning, in some regards similar to what was presented within the concept of problem complexity (Salado & Nilchiani, 2014) or the "Room Theory" (Lipizzi, Borrelli, & Capela, 2020). An example of such a setup is shown in Figure 6.9.

* The content of this section has been extracted literally or with minor editorial modifications from M. Vierlboeck, D. Dunbar, and R. Nilchiani (2022), "Natural Language Processing to Extract Contextual Structure from Requirements," published by IEEE.

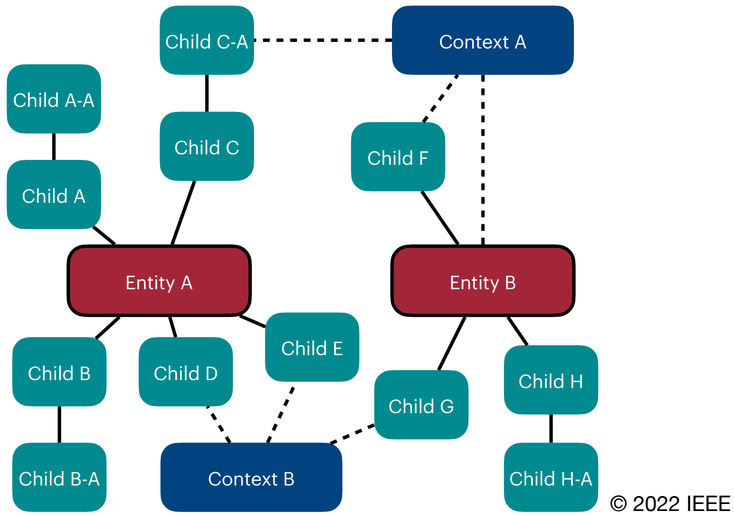

FIG. 6.9 - IMPLICIT & EXPLICIT CONNECTION NETWORK EXAMPLE (VIERLBOECK ET AL., 2022)

For the deduction of these connections, a knowledge base has to be available and tailored to the system since not all context associations apply to all systems. This presented a significant problem for the work in this dissertation since no knowledge base was available for the cases described and studied. Thus, it was decided that this aspect of the approach, and consequently Hypothesis 1 as well, were to be addressed in theory and the possibility proven on a conceptual level. This theoretical construct and the conceptual proof are described below.

Contextual connections can stem from various sources. For instance, different words used or expressions changed can lead to missing links that otherwise should be included and are potentially crucial for the structure. In addition to the usage of different terms, crucial connections can exist that are inferred but not visible on the text layer. An example of these hidden links is requirements that relate to certain aspects of the system without explicitly mentioning said relation. This is in part due to the rigid structure of the statements but also due to human context inference, which is not considered with the explicit text layer processing. Figure 6.10 shows an example of the difference between explicit and implicit connections.

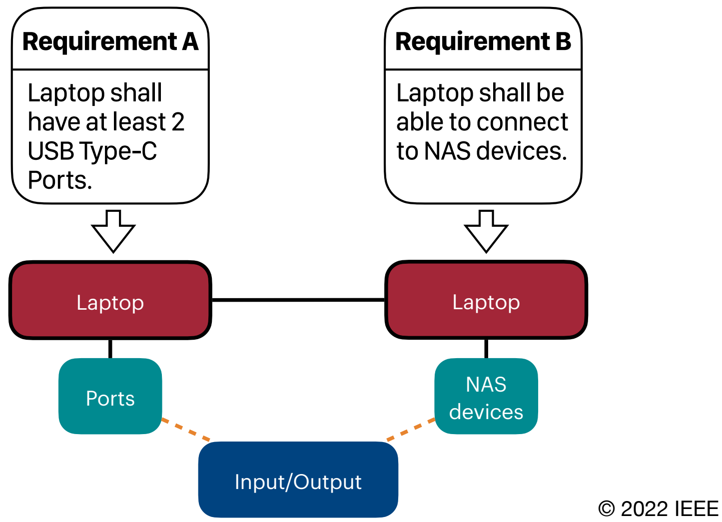

FIG. 6.10 - IMPLICIT CONNECTION ILLUSTRATION (VIERLBOECK ET AL., 2022)

As seen in Figure 6.10, two entities are part of each requirement. Also, since they share the same subject entity, a connection on the top level is derived. Yet, the sentence objects on both sides are different in text and meaning, which does not allow for any link on said level based on explicit information. The two requirements both pertain to the 'Input/Output' capabilities of the system, although they connect to it in a different way. As a result, the context, albeit shared, cannot be elicited on a textual level without inference.

For the provision of contextual information in the form of the knowledge base, an ontology application was chosen due to research parallels and concurrent efforts of the Systems Engineering Research Center (SERC). These parallel efforts allowed for the test and address of Hypothesis 1 without having to create a knowledge base from the ground up. As such, the alignment of a requirement structure and entity identification supported by an ontology is described hereinafter.

In general, ontologies contain a formal representation of knowledge and the relationships between entities, beginning with a subclass taxonomy and expanding over additional relationships such as part_of, describes, and prescribes. Moreover, ontologies can be structured using the Web Ontology Language (OWL), which is based on Description Logics. The use of formal logic allows for automated inference of new knowledge based on existing entities and relationships within the ontology (Sabou, 2016).

For example, a document could list the following requirements:

- “The laptop shall have a solid-state storage device.”
- “The laptop shall have a backup disk drive storage device.”

Assuming another requirement is found elsewhere, reading as follows:

- “The system shall utilize commercial off-the-shelf (COTS) storage devices.”

The last requirement has an inferred relationship with the first two. Since each refers to types of storage devices, the latter requirement puts a constraint on the initial requirements. An ontological representation for this example is presented in Figure 6.11.

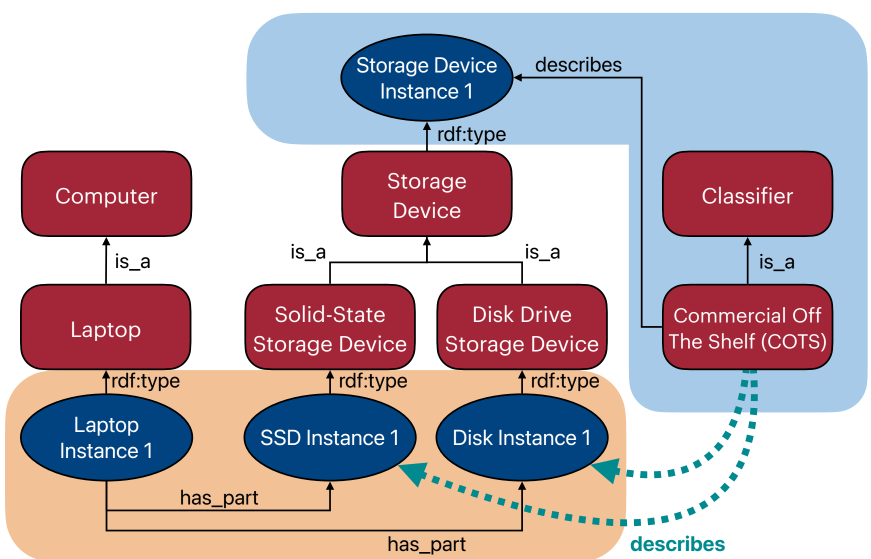

FIG. 6.11 - ONTOLOGICAL REPRESENTATION

The rectangles are classes defined in the domain ontology and show a basic taxonomy. The instances that are mapped from the requirements documents are shown in ovals, and the relationships established by the NLP algorithm are shown. For example, the 'has_part' is the relationship established by the requirements between 'Laptop Instance 1' and 'SSD Instance 1.' From these relationships, a DL reasoner and a rule written in the Semantic Web Rule Language (SWRL) can infer that 'COTS' describes the two storage device instances that are part of the laptop (shown by the dashed line) and as such, provide contextual information that is implicit. These implicit connections are possible due to the existing information of the ontology in combination with the NLP results. This example demonstrates the power that a formal representation of the domain knowledge has when requirements are mapped to an ontology.

Due to the conceptual nature of the approach shown in this section, only function tests in Protégé (Musen, 2015), have been successfully conducted and show the possibility of the implementation as described in the previous paragraphs. The combination with the NLP algorithm was unfortunately not possible due to the size of the existing ontologies that are considered work in progress as of December 2022. Yet, the presented concept and function can be validated through logical reasoning, as outlined in Table 5.2. Reasoning in combination with human checks was thus used in the validation and verification chapter for this part of the approach as well as Hypothesis 1.

To illustrate the possible output under the assumption of an existing ontology, manual inference definitions for the example of the Douglas DC ("DC-1 Request for Proposal Gallery,") are shown in Figure 6.12 based on the results depicted in Figure 6.6. As shown, the entities elicited from the requirements are connected to the contexts of power, performance, and weight based on the overall requirement categories. With the concept shown in this section, these links can be derived from the ontology and thus included automatically.

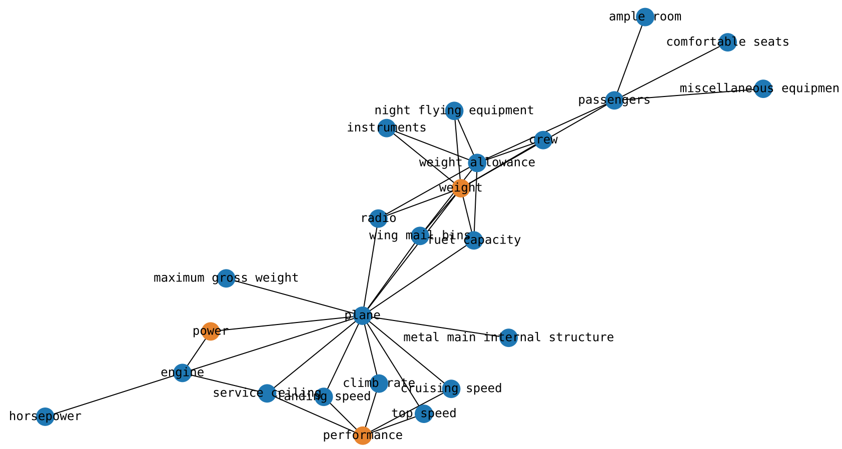

FIG. 6.12 - CREATED NETWORK OF DC PLANE REQ STRUCTURE WITH IMPLICIT CONNECTIONS

Lastly, regarding the scalability of the described concept, we argue that with the existence of a sufficiently large ontology, there are no obstacles for the scalability of the approach and concept since the reasoners can find and elicit connections regardless of the size of the ontology. The only aspect that has to be defined and integrated for scalability is the distance of the connections assessed and thus elicited since a larger ontology would yield many more potential connections and thus the number of false positives increases. Hence, scalability efforts and validity confirmation are pending and considered future research opportunities.

6.6 RESULTING APPROACH SEQUENCE AND USAGE

With the NLP approach, the network/graph creator, and the implicit connection concept, the approach could be applied in the case studies (Chapters 8 & 9). The approach follows the sequential application of the developed parts: 1) the input is processed by the NLP algorithm; 2) the output of the NLP is displayed by the network/graph generator; 3) the results of the network/graph generator are supplemented by the ontology with respective inferences. Since the last step, as described above, depends on the existence of a sufficiently large knowledge base, it had to be omitted for the case studies. The two other steps were fully implemented.

It has to be noted that it would be possible for a human to address the limitations of an insufficient knowledge base manually. By adding the connections based on set rules, a complete implementation would be possible. Yet, this possibility was not pursed for this dissertation due to the focus on removing the necessarily for human supervision/contribution.

To visualize the complete application of the created approach sequence, Figure 6.13 shows the entire process as well as all outputs and results of each step respectively. In total, the approach thus uses the requirements and the ontology as input and yields the entity and from-to-list, an adjacency matrix or equivalent representation, the network graph with all the underlying information, and the extended network graph with the implicit links.

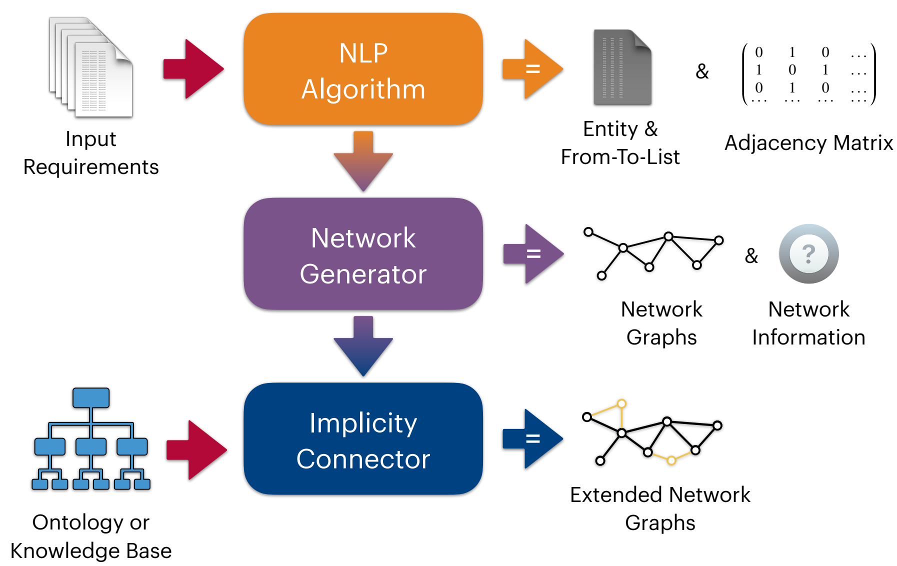

FIG. 6.13 - APPROACH SEQUENCE APPLICATION WITH OUTPUT AND RESULT ELEMENTS

Now, with the approach at hand, the next step, which is tied to the fourth and following hypotheses, could be addressed. Said step was the consideration and qualification of the complexity based on the output of the developments in this chapter. Thus, the next chapter discusses the complexity approach and how it was considered as well as implemented.

Bibliography of Chapter 6

Achour, C. B. (1997). *Linguistic instruments for the integration of scenarios in requirements engineering.* Paper presented at the Proceedings of the Third International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'97), Barcelona, Catalonia.

Choi, J. D., Tetreault, J., & Stent, A. (2015, July). *It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool*, Beijing, China.

Core NLP. (2022). Retrieved from https://https://stanfordnlp.github.io/CoreNLP/

DC-1 Request for Proposal Gallery. Retrieved from http://heroicrelics.org/prologue-rm/request-for-proposal/dscc8864.jpg.html

Ehrlenspiel, K., & Meerkamm, H. (2017). *Integrierte Produktentwicklung* (6th ed.): Carl Hanser Verlag.

GPT-3. (2022). Retrieved from https://openai.com/blog/gpt-3-apps/

IEEE. (1998). IEEE Std 830 - IEEE Recommended Practice for Software Requirements Specifications. New York, NY: Institute of Electrical and Electronics Engineers.

IEEE Guide for Software Requirements Specifications. (1984). *IEEE Std 830-1984*, 1-26. doi:10.1109/IEEESTD.1984.119205

ISO/IEC/IEEE. (2011). ISO/IEC/IEEE International Standard 29148 - Systems and Software Engineering - Life Cycle Processes - Requirements Engineering: ISO/IEC/IEEE.

Jugran, S., Kumar, A., Tyagi, B. S., & Anand, V. (2021). *Extractive Automatic Text Summarization using SpaCy in Python & NLP.* Paper presented at the 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE).

Lipizzi, C., Borrelli, D., & Capela, F. d. O. (2020). A computational model implementing subjectivity with the 'Room Theory'. doi:10.48550/ARXIV.2005.06059

Mohit, B. (2014). Named Entity Recognition. In I. Zitouni (Ed.), *Natural Language Processing of Semitic Languages* (pp. 221-245). Berlin, Germany: Springer Berlin Heidelberg.

Musen, M. A. (2015). The Protégé Project: A Look Back and a Look Forward. *AI Matters, 1*(4), 4–12. doi:10.1145/2757001.2757003

National Aeronautics and Space Administration (NASA). (2020). *NASA Systems Engineering Handbook*. Washington, DC.

Natural Langue Toolkit. (2022). Retrieved from https://www.nltk.org

NetworkX. (2022). Retrieved from https://networkx.org

OpenNLP. (2022). Retrieved from https://opennlp.apache.org

pyvis. (2022). Retrieved from https://github.com/WestHealth/pyvis

Sabou, M. (2016). An Introduction to Semantic Web Technologies. In S. Biffl & M. Sabou (Eds.), *Semantic Web Technologies for Intelligent Engineering Applications* (pp. 53-81). Cham: Springer International Publishing.

Salado, A., & Nilchiani, R. (2014). The Concept of Problem Complexity. *Procedia Computer Science, 28*, 539-546. doi:10.1016/j.procs.2014.03.066

spaCy. (2022). Retrieved from https://spacy.io

spaCy Facts & Figures. (2022). Retrieved from https://spacy.io/usage/facts-figures#benchmarks.

spaCy GitHub page. (2022). Retrieved from https://github.com/explosion/spaCy

Vierlboeck, M., Dunbar, D., & Nilchiani, R. (2022, 25-28 April 2022). *Natural Language Processing to Extract Contextual Structure from Requirements.* Paper presented at the 2022 IEEE International Systems Conference (SysCon).

Vychegzhanin, S., & Kotelnikov, E. (2019). *Comparison of Named Entity Recognition Tools Applied to News Articles.* Paper presented at the 2019 Ivannikov Ispras Open Conference (ISPRAS).

CHAPTER 7: COMPLEXITY QUANTIFICATION OF SYSTEM REQUIREMENTS

"In science we should be interested in things, not persons."

Marie Curie

The structural and topological aspects in Chapter 6 pose the foundation for the quantification of the complexity within requirements, as shown in Figure 5.3. However, the actual quantification method, as well as implications thereof, have yet to be addressed. Thus, this chapter tackles the vital issue of selecting, defining, and putting the quantification approach into context.

Chapter 7 is divided into three sections. The first section (7.1) describes the possible approaches based on the literature shown in Chapter 2 and outlines the choices for the remaining research. Building on the selection, the second section (7.2) illustrates the implementation and use before Section 7.3 provides the interpretation of the application and the effects enabled.

7.1 Selected Approaches and Justification

Looking at the approaches presented in Chapter 2, we see that a multitude of metrics exists to measure and or quantify complexity in general. Yet, not all of these approaches are applicable to the research at hand based on the fact that the problem source and the created NLP approach work with a set of textual requirements and their elicited structure.

Thus, in a first step, the three dimensions of complexity, also discussed in Chapter 2 (Functional, Structural, and Organizational Complexity), were evaluated. Right away, we can rule out the last categories due to the fact that requirements do, for the most part, not address organizational aspects. While organizational requirements exist, they are not part of the main focus that is set forth for this research, as also described in the previous chapters based on the literature. In addition to this exclusion, we argue that the functional aspects are mostly of semantic nature when it comes to textual requirements. This semantic nature also applies in part to the concept of Problem Complexity (Salado & Nilchiani, 2014), for instance. Thus, the inclusion of functional aspects in the research would require upfront interpretation and or definition, which was not chosen to be pursued due to subjectivity. Hence, the structural aspects were the ones chosen for this research. This choice also aligns the research of this dissertation with the works of previous dissertations, such as the ones that resulted in the publications of Salado and Nilchiani (2014), as well as Sinha (2014) and Pugliese (2018).

The focus on structural aspects is also supported by developments (since the early 2000s) in the systems engineering field that have shown a dominance of network and mathematical approaches. This is due to the fact that, in essence, most engineered systems, such as the Internet and GPS, for example, are technical networks (Mahadevan et al., 2006). As a result, the representation of architectures as networks can be used to analyze these structures with graph theoretical approaches. As stated by Crawley et al.: architecture can be defined as an "abstract descriptions of entities [..] and […] relationships" (Crawley et al., 2004). This definition can also be interpreted as a network, which is what the approach presented in Chapter 6 produces. Hence, the networks resulting from the NLP and subsequent steps are structures, and the metrics/quantification processes relying on networks to assess complexity, and other behavioral aspects can be applied as part of the research in this dissertation.

With this choice and first decision, the existing metrics could be assessed and researched, which is provided in Subsection 7.1.1. Subsection 7.1.2 in combination with additional metrics that were chosen based on other criteria that differ from the existing research.

7.1.1 Applicable Structural Complexity Metrics

With the justification and applicability above, metrics that allow for structural complexity assessment are valid contenders to be considered. To assess the metrics, the following can be used: 1) Edmonds (1999), who states the independence of complexity metrics from the observer; 2) McCabe & Butler (1989), who presented a preliminary list of conditions; and 3) Weyuker (1988), who composed a set of nine conditions/properties for complexity metrics.

For completeness' sake, the criteria defined by Weyuker (1988) are summarized here:

1. Different systems shall exist that have different complexity values: $A \neq B$ with $C(A) \neq C(B)$.
2. Only a finite number of systems exist that have a specific complexity value.
3. Different systems shall exist that have the same complexity values: $A \neq B$ with $C(A) = C(B)$.
4. Functionally identical systems shall exist that have different complexity values: $C(A) \neq C(B)$.
5. The complexity value of a union of two systems shall always be greater than the complexity value of the system parts: $C(A \cup B) > C(A)$ and $C(A \cup B) > C(B)$.
6. Two Systems (A & B) with the same complexity values shall be able to yield different compound values when united with the same third System C: $C(A) = C(B)$ with $C(A \cup C) \neq C(B \cup C)$.
7. Two systems shall exist that are permutations of the same components with different complexity value results: $C(A) \neq C(B)$.
8. If a system is a renamed other, they have the same complexity value: $C(A) = C(B)$.
9. Two systems shall exist whose complexity sum is lower than the complexity of their union: $C(A) + C(B) < C(A \cup B)$.

With the above-described criteria, contenders can be evaluated. A comprehensive overview for this evaluation for structural complexity metrics was provided by Sinha (2014) who concluded that only the entropy based approaches in line with 1) and 2) fulfill all nine of Weyuker's criteria. The two metrics that fulfill all criteria were Automorphism-based Entropic Measures (Dehmer, Barbarini, Varmusa, & Graber, 2009) and metrics that calculate the energy E based on the spectrum of a graph (Gutman, 2011; Gutman & Shao, 2011) as shown in Equation 7.1 as a sum of the eigenvalues of the adjacency matrix A.

$$E(A) = \sum_{i=1}^{n} |\lambda_i| \tag{7.1}$$

The Automorphism-based Entropic Measures turned out to be not computable. Hence, the only applicable contenders left, as stated by Sinha (2014), are spectral entropy approaches, which form the core of the Structural Complexity Metric by Sinha & de Weck (2013).

While the implementation and resulting Structural Complexity Metric is valid based on the referenced analysis, it is not compatible with requirements as it relies on system elements and interfaces to calculate the complexity values. Yet, the foundation, being the spectral theory approach, is usable for requirements and thus was chosen as the first metric to apply in the form of Graph Entropy and Laplacian Graph Entropy (LGE), similar to what other researchers applied (Pugliese, 2018). LGE is shown in Equation 7.2 (Gutman & Zhou, 2006), with μ_i being the eigenvalues of the Laplacian Matrix G and e/n being the edges/nodes:

$$E_L(G) = \sum_{i=1}^{n} |\mu_i - \frac{2e}{n}| \tag{7.2}$$

Now, while the selected two metrics would be sufficient and could be applied based on the results that are being enabled by the approach created and shown in Chapter 6, the fact that the foundation of the assessment is requirements prompted a different set of considerations that were also applied to determine potential additional metrics. These considerations are described in the next subsection.

7.1.2 Additional Chosen Metrics

Since requirements are different from actual system elements and components, where the interfaces or the component's internal complexity can be assessed, the selection presented in the previous subsection was extended based on a set of considerations that were made due to the inherent attributes of the results conceptualized in Chapter 6. Mainly, these considerations concerned the connections of requirements regarding their effect on change management and verification/validation in the systems development process.

The first connection and implication to be considered were the one regarding loops within the network. As shown in Figure 6.6, most of the requirements and entities, if simple and not complex, will follow a tree or star structure. Loops in the network would be indicative of a circular connection that connects one entity or one requirement to itself through adjacent connections and nodes. Such circular connections could be potentially problematic for satisfying requirements and or changing them. Let's assume, for example, that three requirements are forming a circle: 1,2, and 3. If Requirement 1 is changed, the change will also affect Requirement 2, which in turn will again affect Requirement 3, closing the loop back to 1. In the best case scenario, the direction of the links does not create a feedback loop that could become reinforcing and make matters worse with each turn, but if a reinforcing behavior emerges, the consequences can cause multiple rework cycles until a solution is found that allows for the satisfaction of all requirements (if such a solution exists at all). Due to this importance of loops, the metric developed by McCabe (1976), called Cyclomatic Complexity, which measures linearly independent loops and or cycles in the architecture, was included despite it not satisfying Weyuker's criteria 2, 6, 7, or 9 (Silva & Kodagoda, 2013).

Cyclomatic Complexity was initially conceptualized by McCabe and Butler under the premise that measuring complexity allows for the connection to cost and time needed to implement a given design (McCabe & Butler, 1989). Furthermore, McCabe and Butler added that complexity assessments shall be conducted before the implementation of the design in order to understand underlying complexity. While initially formulated with a focus on design, we argue that the same connection exists from a requirements perspective and thus include the metrics of Cyclomatic Complexity as shown in Equation 7.3, where e is the number of edges, n is the number of nodes, and P is the number of parts or fragments in the system:

$$C = e - n + 2P \tag{7.3}$$

With the description above, not only was the metric of Cyclomatic Complexity of interest, but also the number of loops by itself as a metric. This measurement is subject to the same drawbacks regarding Weyuker's criteria, but due to the outlined possible dynamics, it was included as a metric to assess. Due to its potential dynamic within the validation/verification process and or change management, this loop count is called 'Load,' or L, to prevent confusion with parts of other metrics. As shown in Equation 7.4, L is the total number of loops l_i in a given network structure.

$$L = |l_i| \tag{7.4}$$

Lastly, a measure of interest often polled against other complexity metrics, and not one that measures complexity directly is the factor of network density. Density d can be assessed only on a network and graph-focused level, with Equations 7.5 below (with e again being the number of edges and n being the number of nodes), or even on an eigenvalue basis as spectral density (Albert & Barabási, 2002), which can even be extended to include Laplacian approaches (Anand & Bianconi, 2009). While a density metric does not fulfill all of Weyuker's criteria above, it can yield some important insights and has been used to measure attributes similar to system complexity in real networks (Elsharief, El-Gawad, & Kim, 2018; Lei, Liu, & Wei, 2019; Leskovec, Kleinberg, & Faloutsos, 2005; Mendling, 2006; Pastor-Satorras, Castellano, Van Mieghem, & Vespignani, 2015; Zou, Su, Qu, & Zhou, 2018). Thus, density was chosen as another factor to look at also over time to see its evolution throughout the development process.

$$d = \frac{e}{Potential\ Connections} \tag{7.5a}$$

$$d = \frac{e}{\frac{n \cdot (n-1)}{2}} \tag{7.5b}$$

In order to model the density over time, another factor that is potentially indicative is the difference or distance of the density of a structure from its minimal density based on the number of nodes currently in the network. A simple structure (Sinha & de Weck, 2013) is either star-shaped or tree-shaped for hierarchies. Such a topology, with the minimum number of edges that does not include fragments, can be seen as the minimum density to create a functioning network. Thus, the metric for the minimum density d_{min} is as follows:

$$d_{min} = \frac{n-1}{\frac{n \cdot (n-1)}{2}} \tag{7.6a}$$

$$d_{min} = \frac{2}{n} \tag{7.6b}$$

By calculating the difference between the actual density of a network and the minimum density, the surplus can be derived that the network has in excess of the minimum possible, which is the next value and metric to look at (see Equation 7.7).

$$d - d_{min} = \frac{e}{\frac{n \cdot (n-1)}{2}} - \frac{2}{n} \tag{7.7}$$

Lastly, one additional factor was considered regarding density. Due to its nature and formula, density is dependent on the size of the network. As seen in equation 7.5b, the actual number of edges is divided by $n^2 - n$. The latter term is exponential in nature and thus will grow faster with increasing network size, yielding an almost inevitable density decline. This decline can be problematic since it introduces the size of the network as a factor that can be hard to consider for comparisons, for instance. Thus, the last aspect to look at was a density approach that adjusts for the size of the network to enable lateral comparisons.

A density approach that accounts for size and thus considers interconnections and betweenness was created and initially used for social networks (Scott, 1988). Recently, said approach was combined with structural entropy (Lei et al., 2019). The approach is called absolute density and is shown in Equation 7.8.

$$d_{absolute} = \frac{e}{\frac{4 \cdot l \cdot R \cdot 3}{3 \cdot D}} \tag{7.8}$$

In equation 7.8, l is the network circumference, r the network radius, and D the diameter. Thus, the formula adjusts for the size by considering the circumference, radius, as well as diameter with the same foundation as 7.5b.

The main reason behind the inclusion of density, as well as its difference from the minimum, was that increasing density indicates a more connected network, which, in conjunction with the thoughts mentioned in this section, may indicate more issues for the verification and validation of requirements as well as rework cycles for change management. Hence, when more connections exist relative to the network size, which means a higher density, the number of problematic links increases, and therefore, we argue that density is a metric of interest to be assessed in the case studies.

In conclusion, the following metrics were chosen as potentially indicative and thus intercity for the case studies:

- Spectral Approaches: Graph Energy and Laplacian Energy
- McCabe/Cyclomatic Complexity
- Load (see Equation 7.4)
- Network Density
- Network Density Delta
- Absolute Network Density

With these metrics, their application was planned in the next section.

7.2 IMPLEMENTATION AND USE

Since the metrics chosen in the last section are geared towards different aspects, which are further explained in the last section of this chapter, a compound metric that units multiple measures into one was considered. For instance, it is possible to combine multiple entropy-based measures into one with the formula, as outlined by Shannon (1948), which is shown in Equation 7.9 below:

$$C(C_1 \cdots C_n) = -\sum_{C_1} \cdots \sum_{C_n} P(c_1 \cdots c_n) \cdot log_j \, [P(c_1 \cdots c_n)] \tag{7.9}$$

With this formula, two insights can be drawn. One, the equation implies that the overall complexity is greater than or equal to the individual values, and thus, each value indicates a minimum. Two, the overall complexity is equal to or lower than the overall sum of the complexities, which indicates a maximum. However, due to the considerations outlined in the previous section, we cannot assume that the entropy-based measures included in the list are complementary in the sense that they are indicative of different aspects of the system/requirements to assess, which is crucial for combination. Thus, the individual assessment was chosen for now without a compound formula.

The same consideration was made for the remaining four metrics and measures. Since all of them target different insights and implications, as per the next section, a combination could not be derived in a logical or mathematical way. Furthermore, since the actual effects of the metrics had to be first tested in case studies, the compounding or combination would have potentially corrupted or worsened the results of the studies. Thus, individual assessments and interpretation were chosen for all metrics in the frame of this research.

This choice means that the metrics chosen and outlined in this chapter were applied in the case studies to assess their suitability and indication power in line with the hypotheses set forth for this research. This way, the actual implications of each metric can be assessed first, and in a subsequent step, compound and more comprehensive metrics can be considered. Thus, the defined and chosen measures are directly applied and demonstrated holistically in the case studies (see Chapter 8 for a demonstration and Chapter 10 for implication assessment).

7.3 Interpretation, Planned Insights, and Effects

The last aspect to discuss in this chapter, based on the decision to conduct individual assessments for the metrics, are the implications and planned insights to achieve with the metrics. Thus, this last section includes an overview of the metrics listed, also considering the different layers for the analysis shown in Figure 6.7.

7.3.1 Graph Energy and Laplacian Graph Energy

The first two metrics to apply and test are directly related to research mentioned in the literature review as well as earlier in this chapter. Said research, targeting structural complexity, has already been shown to apply to the design and or architecture of a system with different effects. For instance, Sinha (2014) relates the contributions for structural complexity to Bellman's (2011), which connects complex system design to development effort and cost (Sinha & de Weck, 2016). Similar links have been shown in even earlier publications as well (MacCormack, Rusnak, & Baldwin, 2006). In close relation to the source data of the third case study, Pugliese (2018) also evaluated connections between spectral metrics and effort.

Now, given the existing research above, the application in the context of this dissertation targets the assumption that the metrics of Graph Energy and or Laplacian Energy will also have an effect on the system development process, but not through the architecture or design of the system. Thus, the effect of the requirements can be assessed. This thought and consideration process is shown in Figure 7.1. As seen, the application of the metrics shall examine if the entropy within the requirements or even on the entity level can be used to gauge the effort necessary that is needed to work with, create, integrate, or verify a given set of requirements. As a result, comparison between different constellations could also be enabled based on the metrics.

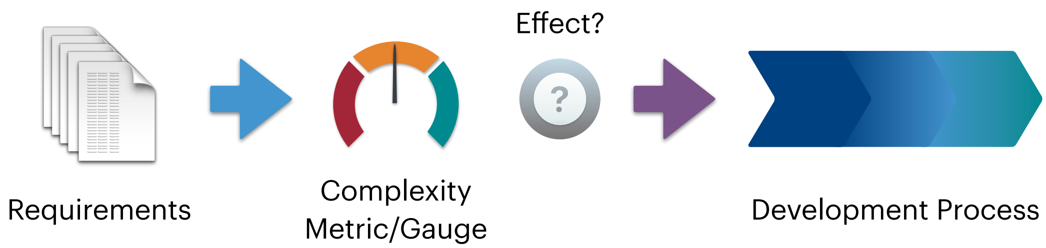

Fig. 7.1 - Causal Effect Chain of Requirements on the Development Process

If applicable, by evaluating a set of requirements, the direct effect on the effort and resulting cost of the process can be gauged and potentially also included in the budgeting process, as recommended by Nilchiani and Pugliese (2017) for instance.

7.3.2 McCabe/Cyclomatic Complexity

Similar to the spectral metrics and approaches above, McCabe/Cyclomatic Complexity has been related to the cost and time/effort in projects and developments, predominantly in the software engineering space due to its origin (Basili & Perricone, 1984; Bhansali, 2005; Chen, 1978; Gill & Kemerer, 1991; Hirota, Tohki, Overstreet, Hashimoto, & Cherinka, 1994; Honglei, Wei, & Yanan, 2009; Kushwaha & Misra, 2008; T. J. McCabe, 1996; Mende & Koschke, 2010; Rana, Khan, & Shamail, 2006; Sturtevant, 2013; Subandri & Sarno, 2017; Woodfield, Shen, & Dunsmore, 1981). Yet, due to the foundation of Cyclomatic Complexity being concerned with loops and fragments/parts and thus defining logical complexity, which also affects requirements (Egyed, Graf, & Grünbacher, 2010), we argue that these points apply to the research at hand as well, and not just software engineering. Furthermore, information paths are an integral aspect of Cyclomatic Complexity, which also is critical when it comes to aspects like effort and cost. Thus, the causal chain shown in Figure 7.1 applies to this metric as well and will be demonstrated/tested in the case studies to determine the impact and effects.

7.3.3 Load

The Load, being defined as a new metric in this dissertation, as shown in Equation 7.4, is a hybrid between the Cyclomatic Complexity above and the density approaches below. Due to its inclusion of loops as a measure, it directly targets the measure of circular connections and potential rework cycles (Foreman, Moigne, & de Weck; Wertz, Everett, & Puschell, 2011)(also see Figure 7.2), which are related to increased cost in change management (Boznak, 1994; Boznak & Decker, 1993; Lindemann & Reichwald, 1998) due to the efforts necessary for satisfying/verifying requirements in a loop. Furthermore, loops can also affect the overall risk of the development process, and it is strongly recommended to be considered loops as such (National Aeronautics and Space Administration (NASA), 2020a, 2020b). Hence, while the effect of loops has to be empirically investigated, their existence and connected interfaces are a first step in that direction, which was evaluated as part of this dissertation.

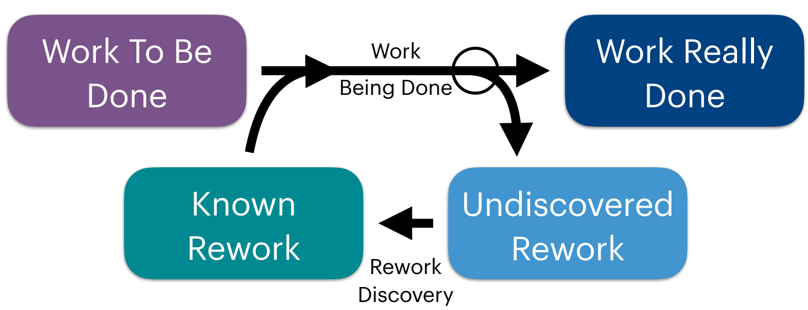

Fig. 7.2 - The Rework Cycle (Cooper, 1994)

7.3.4 Density, Density Delta, and Absolute Density

Lastly, the density metrics are the extension of the Load and target the overall connectedness of the network. As shown by multiple studies and concepts, density overall can be an indicator for a variety of aspects in a network, ranging from errors in models (Leskovec et al., 2005), to defect impacts (Sturtevant, 2013), system resilience (Gao, Barzel, & Barabási, 2016; Goryashko, Samokhine, & Bocharov, 2019), efficiency effects (Strang, Haynes, Cahill, &

Narayan, 2018), local importance effects (Hui Xu & Lun, 2019), economic factors (Battiston, Caldarelli, May, Roukny, & Stiglitz, 2016; Hearnshaw & Wilson, 2013; Isik, 2011; Mizutani & Urakami, 2001), and even extended complexity metric aspects (Lei et al., 2019).

Hence, we argue that the density of the network can have an impact on the development process by influencing the effort of the work necessary to deal with a set of requirements, similar to the implications of the previous metrics. While it is difficult to identify the direct impact, the overall effect is of interest due to the emergent properties of the complexity problems.

The density of the requirements can affect different aspects of the RE process as well, ranging from selection, handling, management, all the way to verification, due to the above-mentioned possible implications. For example, the phenomena shown by Strand et al. (2018) could have direct implications for the change management applications since local efficiencies in clusters could indicate easier adjustments within a specific area of a requirement set. Therefore, such effects, while not directly connected in a causal way to the metrics presented, are of interest and were evaluated and tested.

With all the metrics and their reasoning outlined, the actual case studies could be addressed. This is described in Chapters 8 through 10 and outlines both the developed approach as well as the metrics and their implications.

Bibliography of Chapter 7

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics, 74*(1), 47-97. doi:10.1103/RevModPhys.74.47

Anand, K., & Bianconi, G. (2009). Entropy measures for networks: Toward an information theory of complex topologies. *Physical Review E, 80*(4), 045102. doi:10.1103/PhysRevE.80.045102

Basili, V. R., & Perricone, B. T. (1984). Software errors and complexity: an empirical investigation. *Communications of the ACM, 27*(1), 42-52.

Battiston, S., Caldarelli, G., May, R. M., Roukny, T., & Stiglitz, J. E. (2016). The price of complexity in financial networks. *Proceedings of the National Academy of Sciences, 113*(36), 10031-10036. doi:doi:10.1073/pnas.1521573113

Bellman, K. (2011). *Model-Based Design, Engineering, and Development: Advancements mean New Opportunities for Space System Development*. Paper presented at the AIAA SPACE 2011 Conference & Exposition. https://arc.aiaa.org/doi/abs/10.2514/6.2011-7304

Bhansali, P. V. (2005). Complexity measurement of data and control flow. *ACM SIGSOFT Software Engineering Notes, 30*(1), 1.

Boznak, R. G. (1994). When doing it right the first time is not enough. *Quality Progress, 27*(7), 74-78.

Boznak, R. G., & Decker, A. K. (1993). *Competitive Prod Dev: A Quality Approach to Succeeding in the 90s and beyond*. New York, NY: McGraw-Hill Education.

Chen, E. T. (1978). Program Complexity and Programmer Productivity. *IEEE Transactions on Software Engineering, SE-4*(3), 187-194. doi:10.1109/TSE.1978.231497

Cooper, K. G. (1994). The $2,000 hour: How managers influence project performance through the rework cycle. *Project Management Journal, 25*, 11-11.

Crawley, E., De Weck, O., Magee, C., Moses, J., Seering, W., Schindall, J., . . . Whitney, D. (2004). The influence of architecture in engineering systems. *Engineering Systems Monograph*.

Dehmer, M., Barbarini, N., Varmusa, K., & Graber, A. (2009). A Large Scale Analysis of Information-Theoretic Network Complexity Measures Using Chemical Structures. *PLoS ONE, 4*(12).

Edmonds, B. (1999). *Syntactic measures of complexity*: The University of Manchester (United Kingdom).

Egyed, A., Graf, F., & Grünbacher, P. (2010, 27 Sept.-1 Oct. 2010). *Effort and Quality of Recovering Requirements-to-Code Traces: Two Exploratory Experiments.* Paper presented at the 2010 18th IEEE International Requirements Engineering Conference.

Elsharief, M., El-Gawad, M. A. A., & Kim, H. (2018). Density Table-Based Synchronization for Multi-Hop Wireless Sensor Networks. *IEEE Access, 6*, 1940-1953. doi:10.1109/ACCESS.2017.2780923

Foreman, V. L., Moigne, J. L., & de Weck, O. A Survey of Cost Estimating Methodologies for Distributed Spacecraft Missions. In *AIAA SPACE 2016*.

Gao, J., Barzel, B., & Barabási, A.-L. (2016). Universal resilience patterns in complex networks. *Nature, 530*(7590), 307-312. doi:10.1038/nature16948

Gill, G. K., & Kemerer, C. F. (1991). Cyclomatic complexity density and software maintenance productivity. *IEEE Transactions on Software Engineering, 17*(12), 1284-1288. doi:10.1109/32.106988

Goryashko, A., Samokhine, L., & Bocharov, P. (2019). About complexity of complex networks. *Applied Network Science, 4*(1), 87. doi:10.1007/s41109-019-0217-1

Gutman, I. (2011). *Hyperenergetic and hypoenergetic graph*: Math. Inst., Belgrade.

Gutman, I., & Shao, J.-Y. (2011). The energy change of weighted graphs. *Linear Algebra and its Applications, 435*(10), 2425-2431. doi:10.1016/j.laa.2011.02.045

Gutman, I., & Zhou, B. (2006). Laplacian energy of a graph. *Linear Algebra and its Applications, 414*(1), 29-37. doi:10.1016/j.laa.2005.09.008

Hearnshaw, E. J. S., & Wilson, M. M. J. (2013). A complex network approach to supply chain network theory. *International Journal of Operations & Production Management, 33*(4), 442-469. doi:10.1108/01443571311307343

Hirota, T., Tohki, M., Overstreet, C. M., Hashimoto, M., & Cherinka, R. (1994, 7-9 Dec. 1994). *An approach to predict software maintenance cost based on ripple complexity.* Paper presented at the Proceedings of 1st Asia-Pacific Software Engineering Conference.

Honglei, T., Wei, S., & Yanan, Z. (2009, 25-27 Dec. 2009). *The Research on Software Metrics and Software Complexity Metrics.* Paper presented at the 2009 International Forum on Computer Science-Technology and Applications.

Hui Xu, J. Z., Jing Yang,, & Lun, L. (2019). Node Importance Ranking of Complex Network based on Degree and Network Density. *Int J Performability Eng, 15*(3), 850-860. doi:10.23940/ijpe.19.03.p14.850860

Isik, F. (2011). Complexity in Supply Chains: A New Approach to Quantitative Measurement of the Supply-Chain-Complexity. In P. Li (Ed.), *Supply chain management* (Vol. 21, pp. 417-432): INTECH Open Access Publisher.

Kushwaha, D. S., & Misra, A. K. (2008). Software test effort estimation. *ACM SIGSOFT Software Engineering Notes, 33*(3), 1-5.

Lei, M., Liu, L., & Wei, D. (2019). An Improved Method for Measuring the Complexity in Complex Networks Based on Structure Entropy. *IEEE Access, 7*, 159190-159198. doi:10.1109/ACCESS.2019.2950691

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2005). *Graphs over time: densification laws, shrinking diameters and possible explanations.* Paper presented at the Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD '05).

Lindemann, U., & Reichwald, R. (1998). *Integriertes Änderungsmanagement*. Berlin Heidelberg, Germany: Springer-Verlag.

MacCormack, A., Rusnak, J., & Baldwin, C. Y. (2006). Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code. *Management Science, 52*(7), 1015-1030. doi:10.1287/mnsc.1060.0552

Mahadevan, P., Krioukov, D., Fomenkov, M., Dimitropoulos, X., Claffy, K., & Vahdat, A. (2006). The Internet AS-level topology: three data sources and one definitive metric. *ACM SIGCOMM Computer Communication Review, 36*(1), 17-26. doi:10.1145/1111322.1111328

McCabe, T. J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering, SE-2*(4), 308-320. doi:10.1109/TSE.1976.233837

McCabe, T. J. (1996). Cyclomatic complexity and the year 2000. *IEEE Software, 13*(3), 115-117. doi:10.1109/52.493032

McCabe, T. J., & Butler, C. W. (1989). Design complexity measurement and testing. *Communications of the ACM, 32*(12), 1415-1425. doi:10.1145/76380.76382

Mende, T., & Koschke, R. (2010, 15-18 March 2010). *Effort-Aware Defect Prediction Models.* Paper presented at the 2010 14th European Conference on Software Maintenance and Reengineering.

Mendling, J. (2006). *Testing Density as a Complexity Metric for EPCs*.

Mizutani, F., & Urakami, T. (2001). Identifying network density and scale economies for Japanese water supply organizations*. *Papers in Regional Science, 80*(2), 211-230. doi:10.1111/j.1435-5597.2001.tb01795.x

National Aeronautics and Space Administration (NASA). (2020a). *NASA Cost Estimating Handbook* (4th ed.). Washington, DC.

National Aeronautics and Space Administration (NASA). (2020b). *NASA Systems Engineering Handbook*. Washington, DC.

Nilchiani, R., & Pugliese, A. (2017). *A Systems Complexity-based Assessment of Risk in Acquisition and Development Programs*. Retrieved from https://dair.nps.edu/handle/123456789/2689

Pastor-Satorras, R., Castellano, C., Van Mieghem, P., & Vespignani, A. (2015). Epidemic processes in complex networks. *Reviews of Modern Physics, 87*(3), 925-979. doi:10.1103/RevModPhys.87.925

Pugliese, A. (2018). *Development of Spectral Structural Complexity Metrics in Cyber-Physical Systems.* (Doctor of Philosophy). Stevens Institute of Technology, Hoboken, NJ.

Rana, Z. A., Khan, M. J., & Shamail, S. (2006, 13-14 Nov. 2006). *A comparative study of spatial complexity metrics and their impact on maintenance effort.* Paper presented at the 2006 International Conference on Emerging Technologies.

Salado, A., & Nilchiani, R. (2014). The Concept of Problem Complexity. *Procedia Computer Science, 28*, 539-546. doi:https://doi.org/10.1016/j.procs.2014.03.066

Scott, J. (1988). Social Network Analysis. *Sociology, 22*(1), 109-127. doi:10.1177/0038038588022001007

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*(3), 379-423. doi:10.1002/j.1538-7305.1948.tb01338.x

Silva, D. I. D., & Kodagoda, N. (2013, 26-28 April 2013). *Applicability of Weyuker's properties using three complexity metrics.* Paper presented at the 2013 8th International Conference on Computer Science & Education.

Sinha, K. (2014). *Structural complexity and its implications for design of cyber-physical systems.* Massachusetts Institute of Technology,

Sinha, K., & de Weck, O. (2013). *A network-based structural complexity metric for engineered complex systems.* Paper presented at the 2013 IEEE International Systems Conference (SysCon).

Sinha, K., & de Weck, O. (2016). Empirical Validation of Structural Complexity Metric and Complexity Management for Engineering Systems. *Systems Engineering, 19*(3), 193-206. doi:10.1002/sys.21356

Strang, A., Haynes, O., Cahill, N. D., & Narayan, D. A. (2018). Generalized relationships between characteristic path length, efficiency, clustering coefficients, and density. *Social Network Analysis and Mining, 8*(1), 14. doi:10.1007/s13278-018-0492-3

Sturtevant, D. J. (2013). *System Design and the Cost of Architectural Complexity.* Massachusetts Institute of Technology,

Subandri, M. A., & Sarno, R. (2017). Cyclomatic Complexity for Determining Product Complexity Level in COCOMO II. *Procedia Computer Science, 124*, 478-486. doi:10.1016/j.procs.2017.12.180

Wertz, J. R., Everett, D. F., & Puschell, J. J. (2011). *Space Mission Engineering - The New SMAD*: Microcosm Press.

Weyuker, E. J. (1988). Evaluating software complexity measures. *IEEE Transactions on Software Engineering, 14*(9), 1357-1365. doi:10.1109/32.6178

Woodfield, S. N., Shen, V. Y., & Dunsmore, H. E. (1981). A study of several metrics for programming effort. *Journal of Systems and Software, 2*(2), 97-103. doi:10.1016/0164-1212(81)90029-7

Zou, Z., Su, X., Qu, X., & Zhou, P. (2018). DA-Net: Learning the Fine-Grained Density Distribution With Deformation Aggregation Network. *IEEE Access, 6*, 60745-60756. doi:10.1109/ACCESS.2018.2875495

CHAPTER 8: CASE STUDY 1 - SKYZER (UNMANNED AERIAL VEHICLE)

"The true method of knowledge is experiment."

William Blake

The case studies were conducted with the foundation and approaches provided in the last two chapters. In total, three case studies were used to verify and address the hypotheses, in addition to the information already described in Section 6.5, which addressed Hypotheses 3. The first case study targets Hypotheses 1 & 2. For this purpose, an ongoing project from the Systems Engineering Research Center (SERC) was used. This project deals with Model-Based Systems Engineering (MBSE) topics and contains an unmanned aerial vehicle (UAV) as an exemplary development subject. To show the application of the developed approaches and results, this chapter is divided into four sections. The first section describes the content, circumstances of the case study, and the UAV (8.1). Section 8.2 describes the application and process of the case study. The third section presents the results and shows different representations (8.3). The fourth section (8.4) discusses the outcome and implications before the results verification (8.5) concludes this chapter.

8.1 Case Study Content and Circumstances

As described above, the Skyzer UAV is a hypothetical and experimental project to test and develop various new tools, such as cost, requirements management, and interoperability (Ballard et al., 2020). Within this project, different aspects of the development and acquisition process are addressed and explored, such as the Mission Model, System Model, and Request for Proposal Document. The concept of operations (CONOPS) for the Skyzer UAV is that the system provides humanitarian maritime support. Figure 8.1 shows an exemplary graphical representation of the CONOPS (Blackburn, 2018).

FIG. 8.1 - GRAPHICAL CONOPS FOR SKYZER UAV (BLACKBURN, 2018)

The experimental development process of Skyzer also contained a complete set of requirements for one of the subsystems of the UAV: the landing gear and interfacing components. This specification document was used for the first case study. The initial document of the landing gear contained 49 requirements in an Excel file. The requirements were named and numbered but did not contain a hierarchy beyond their identifiers. For the case study, the requirement text was used and assessed for quality in the first step. In this step, it was discovered that the requirements were not entirely written in accordance with the standards or the recommendations described in Chapter 3. Thus, to pre-process the data, the requirements were improved to adhere to the 'shall'/'will' statement structure. This also included separating requirements that had been combined due to similarity, which also helped resolve ambiguities caused by said combinations. After the improvements, 79 individual requirements were available for the Skyzer landing gear, which were then used for the case study in this chapter. The full list of the requirements can be found in Appendix C. Note that despite the improvements, the requirements were not optimal, and in some cases, the values of certain attributes were still 'to be defined'/'TBD.' This is considered in Section 8.4.

8.2 Approach Application and Process

The implementation of the NLP and network/graph approach described in Chapter 6 was conducted with the 79 requirements. Unfortunately, no complete or sufficient ontology existed at the time of this case study in early 2022 to allow for the implementation of a knowledge base and subsequent definition of implicit connections. The entire case study was created and conducted in a Jupyter ("Jupyter," 2023) environment, which ensured version control and export possibilities as IPython Notebook files. The use of Jupyter notebooks also allowed for easy segmentation of the code and implementation of the quantification analysis approaches.

The entire process and code relied on various open-source libraries, such as pandas ("pandas," 2022), for the structuring of tabular data. A complete list of the used libraries, as well as the source-code, can be provided upon request. Figure 8.2 shows the outlined sequence's individual steps and outputs, as well as processed information.

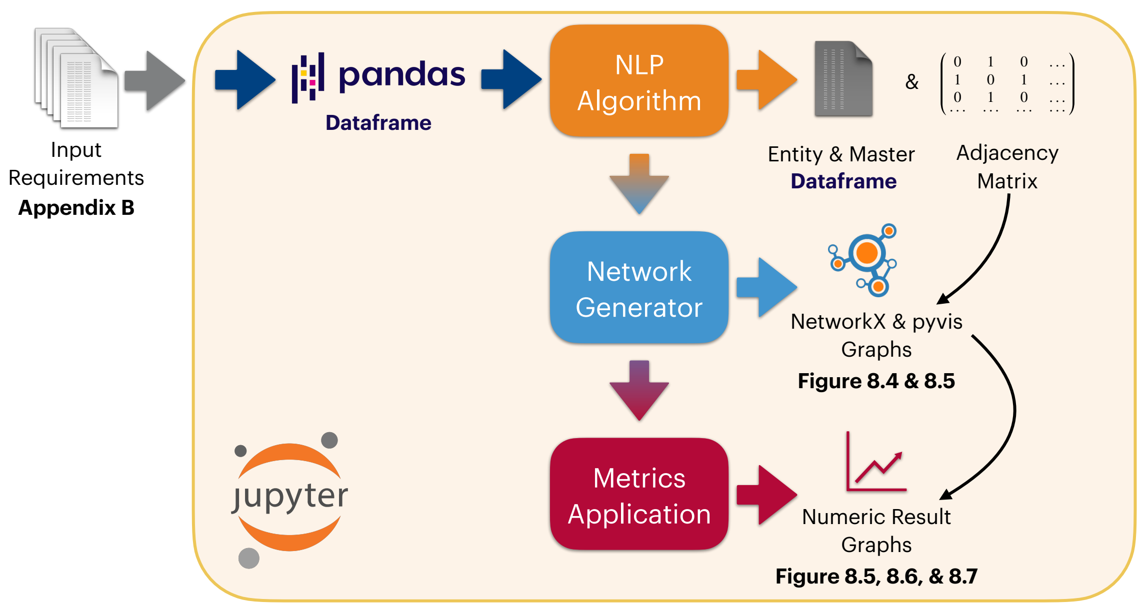

FIG. 8.2 - PROCESS OVERVIEW

As shown in the figure above, the first step was to import the requirements document. For this step, the document was left unchanged and transferred directly from the Excel format into an identically structured pandas data frame. This direct transfer ensured the retention of all the textual information. With this frame, the column of the requirement text was first split into individual sentences, which yielded the list of requirements. If future sets contain structural information in the form of IDs, a dictionary approach instead of a list is recommended. The list of requirements was subsequently used as an input for the NLP sequence shown in Figure 6.5. The result of the process was the list containing all the entities and their from-to connections. Furthermore, the NLP process yielded a list of all individual entities and a list that outlined which entity was part of which requirement. The list describing the associations of the entities and their sources represents the content of the requirements. Lastly, based on the content of the requirements, a list similar to the from-to list of the entities was created for the individual requirements. This last list was the foundation for the requirement structure built on top of the entity/NLP structure (see Figure 6.7).

The lists resulting from the NLP approach were then processed with the NetworkX ("NetworkX," 2022) library to generate networks and graph representation based on the from-to lists. The output and NetworkX graphs could subsequently be directly fed into the pyvis ("pyvis," 2022) library to generate an interactive representation that could be manipulated for visual analysis. Furthermore, the NetworkX graph foundations were used for the calculations and creation of adjacency matrices as well as various other attributes, such as fragment count, edge count, node count, and density. The graphical representations, numerical results, and first analyses that were conducted are shown in the next section and discussed in 8.4. In addition, as a foundation for the correctness of the results, the necessary metrics used in the validation chapter are explained, including the description of their origins.

8.3 Results

To structure the results in accordance with the layers of the approach (hierarchy, requirement structure, and NLP structure - Figure 6.8), this section is divided accordingly in addition to the numerical results obtained through the application of selected metrics.

8.3.1 Entity and NLP Structure Results

Regarding the entity and NLP structure results, the requirements of the Skyzer UAV (attached in Appendix C) were processed individually, and the output was recorded in a CSV (comma-separated values) file with three columns: the entity from which the connection originated, the entity that was receiving the connection, and the number of the requirement that contained the entities. Also, a list with all individual entities that were identified was generated.

The table with the connections turned out to be 329 lines long, which means that there are 329 connections in the requirement document. While this does not necessarily indicate a network with 329 edges, it is representative of the number of references/links in the document. After removing duplicate terms, the result showed that there were 252 individual entities in the document. These entities had a total of 265 edges after accounting for identical connections that appeared more than once. In addition, the network turned out not to be entirely connected and instead showed 12 fragments, of which 11 had less than ten nodes. Thus, a main network structure existed with small fragments that were separated from it. Further manual analysis showed that some of the fragments were separated from the main network due to terms that did not occur entirely in another requirement, such as 'landing gear down locks,' for example, which could be linked to 'landing gear,' but due to them not being identical, the graph creation algorithm treats them separately. Figure 8.3 shows a plot of the entity network.

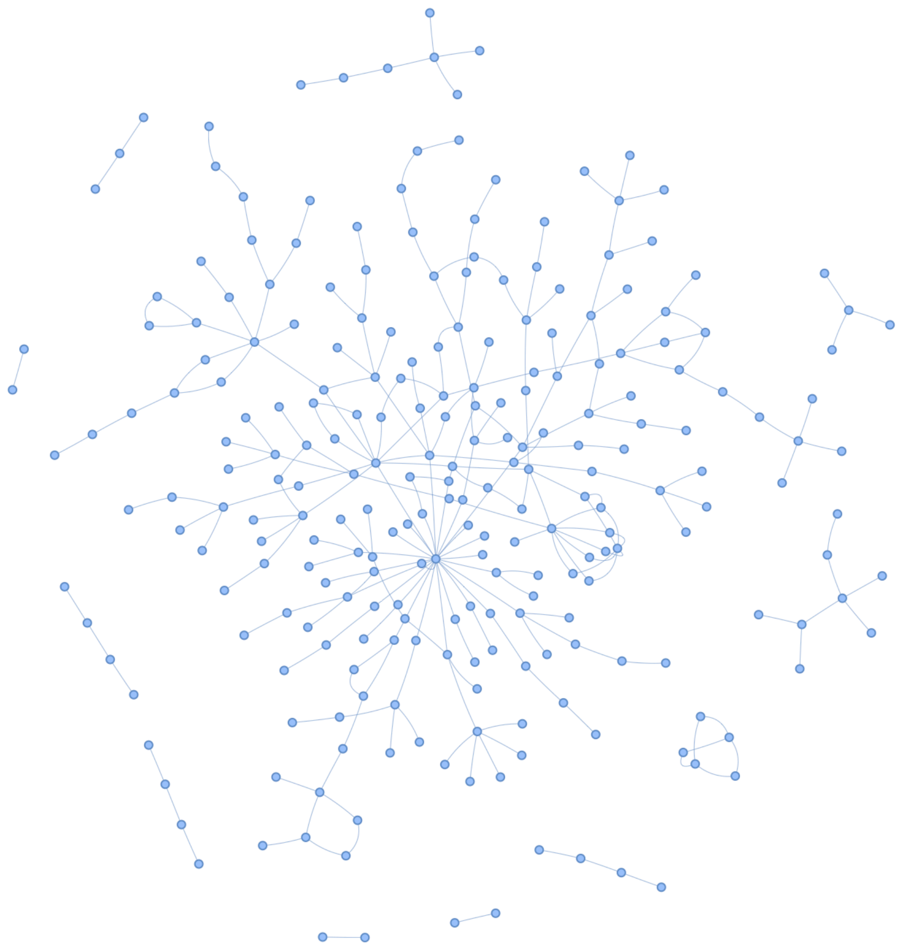

FIG. 8.3 - ENTITY STRUCTURE NETWORK PLOT

It has to be noted that the removal of duplicate references might not be required or useful in the future since the number of edges between two nodes might be a metric worth analyzing. Due to the standard graph structure and the metrics described in Chapter 7 relying on singular-edge networks, the decision to remove them was made for this case study.

As described in the approach section, the requirement structure could be derived from the entity structure by using the connections on the level above the entities. These results are described in the next section.

8.3.2 Requirement Structure Results

Since the requirement structure does not require as much elicitation as the NLP/entity structure, the results were more predictable and, to some extent, usable for verification right away. For instance, the network of the requirements had 79 nodes, which represents the number of requirements in the initial document. The number of edges in the requirement structure was significantly higher than the one of the entities with 373 links. This higher number is due to the fact that the entity structure counts and depicts an entity only as one node, but within the requirement structure, the entity and thus, all of its connections can be part of more than one requirement, which increases the number of links.

Figure 8.4 depicts the requirement structure with color-coded edges and nodes. Nodes that are not filled have no edges and were mentioned once in the adjacency list; **light grey nodes** have fewer than 12 mentions; **dark grey nodes** have more than 12 but fewer than 25 mentions; **light blue nodes** have more than 25 but fewer than 50 mentions; and **dark blue nodes** appeared more than 50 times.

Looking at the fragments and parts of the requirements structure, we see a similar picture with some fragments scattered around an integrated main part. The fragments and unconnected networks have been manually checked and linked to the fragments seen in 8.3, further confirming the approach. The higher density in the core, compared to the entity structure, comes from the fact that terms such as the 'landing gear' appear as one node in the entity structure but end up connecting all nodes that contain said term in the requirement structure. This is important to keep in mind as the analysis can thus help and target different efforts. For instance, the requirement structure can inform about RE tasks, whereas the NLP and entity structure can be used with architectural aspects in mind.

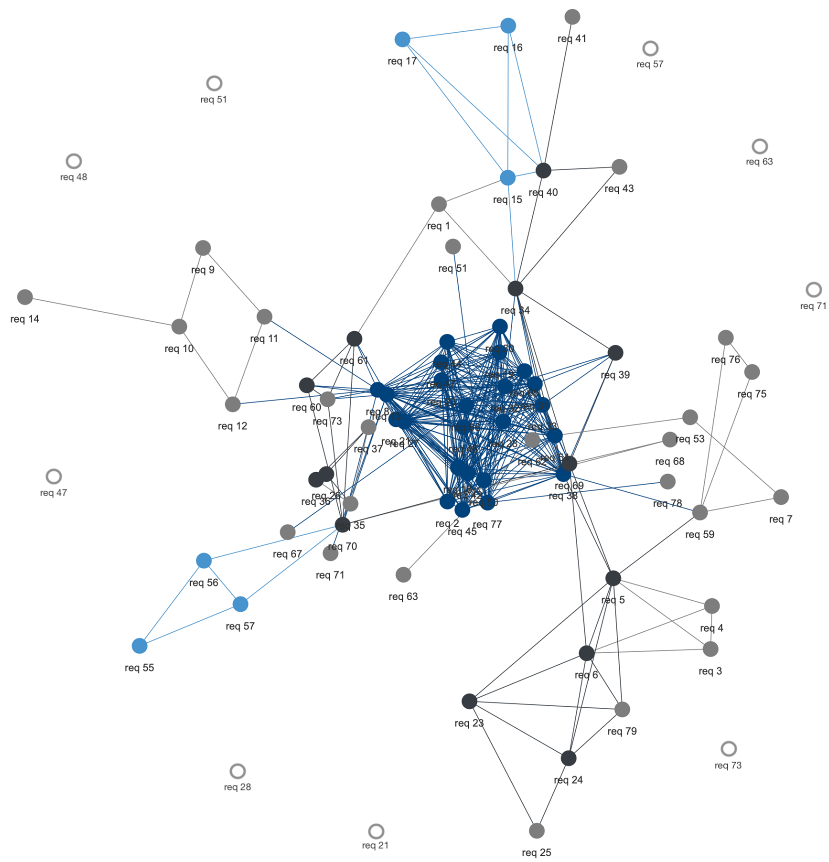

FIG. 8.4 - REQUIREMENT STRUCTURE NETWORK PLOT

Now, with the requirement and entity structure generated, the numerical aspects of the results could be assessed, which is described in the next section.

8.3.3 NUMERICAL AND METRIC RESULTS

Based on the results above, specific metrics could be analyzed and looked at. For this chapter, since additional work in this direction in an organized case study is provided in Chapter 10, only a select number of numeric results are looked at that were of interest at the time of the case study, as well as due to its origins at the System Engineering Research Center. Thus, the following metrics are presented: density and density delta of the requirement network (including time dependency), Graph Energy in the form of entropy, and Cyclomatic Complexity (McCabe, 1976) due to its inclusion of fragment/part count as well as information paths. These select metrics are also in line with the methods chosen in Chapter 7.

First, as already alluded to in the previous subsection, the density numbers of the two networks differ greatly. The entity/NLP structure has a density of 0.00858, whereas the requirement network achieves a density of 0.121. This is significant and mostly caused by the way that the two networks are constructed, with the dense main network in the center of the requirements structure, as seen in Figure 8.4.

Looking at the density of the requirement structure in accordance with their chronology/addition order (used instead of a hierarchy for this case study) reveals its progression when one requirement after the other is added; this is shown in Figure 8.5. As seen in the chart, the density first stays high due to the fact that each requirement added is connected to the previous one(s), and the number of possible connections is low (see exponential denominator in Equation 7.5b). Soon though, the density drops since the number of possible edges exceeds the actual connections due to its exponential origin. Then, after about a dozen requirements, the reduction slows and, for some requirements, even turns into an increase until it finally reaches its end value. Also plotted in the graph as a reference is the minimum density possible with the given number of edges since this value would be associated with a star or tree structure, which would be considered easy and less complex from a structural and topological perspective (Sinha & de Weck, 2013). Thus, the difference between the minimum density and the actual density is of potential interest and is also discussed in Chapter 7 and Section 8.4 below.

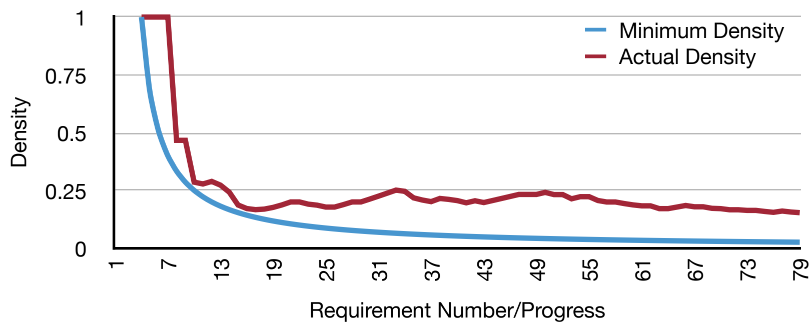

FIG. 8.5 - REQUIREMENT STRUCTURE DENSITY PROGRESS

Second, for the Graph Energy, the same approach was applied as for the requirements, but this time for the entity/NLP structure as a calculation basis. Thus, the progress of the entropy in the entity structure can be analyzed through the progress of the requirement addition process. The resulting graph is shown in Figure 8.6. As the figure depicts, a steady increase in entropy was recorded throughout the process. What is interesting and worth noting is that the impact of the individual requirements is not equal and ranges from 0 to 10.5 with an average of 3.6. This shows that the effect of the individual requirements differs and can be of interest for various steps in the RE and development process. These effects are discussed in Section 8.4 in more detail.

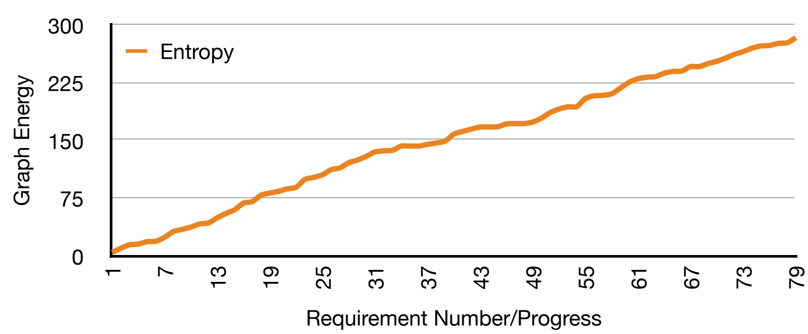

FIG. 8.6 - REQUIREMENT STRUCTURE GRAPH ENERGY PROGRESS

Lastly, for the Cyclomatic Complexity of the case study data, which includes the number of edges, nodes, and parts (also sometimes described as fragments), Figure 8.7 shows a similar picture to the Graph Energy as entropy, but with a much less steady profile. The slow increase in the beginning is due to the even number of edges and nodes added, but once about 20 requirements are introduced, the cross-connections add significantly more edges than nodes, leading to an increase. Also, the sudden increases indicate highly connected requirements, which match the colors in Figure 8.4. Thus, Cyclomatic Complexity adds another layer of transparency to the results and can be interpreted respectively as well.

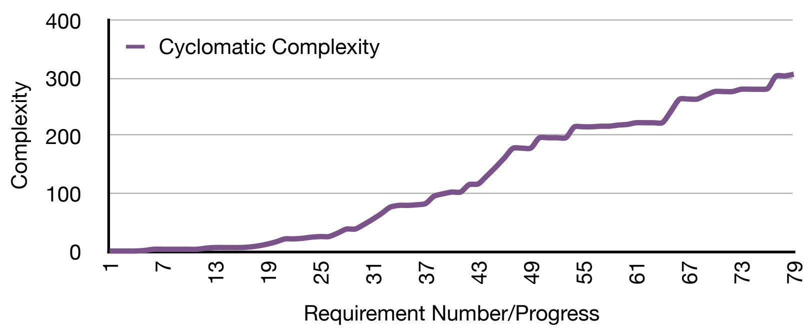

FIG. 8.7 - REQUIREMENT STRUCTURE CYCLOMATIC COMPLEXITY PROGRESS

Now, with these numeric results, which are further expanded and analyzed in more detail in the third case study, a discussion is possible to look into the possibilities and insights that the developed approach provides. This discussion is described in the last section below.

8.4 DISCUSSION AND IMPLICATIONS

In order to discuss the results and insights gained from the Skyzer case study in an organized fashion that matches the order of the results, three parts are discussed: 1) the overall structures and on a graph level; 2) anomalies in the graphs and fragments; and 3) insights that can be deduced from the metric and numerical results.

8.4.1 Overall Structures

Looking at the overall structures that resulted from the case study (Figures 8.3 and 8.4), we see that for the entity structure (8.4), the overall shape resembles a star or tree. This is to be expected since the overall structure is a decomposition of the entities in the system and thus follows a breakdown order. The center node is the 'landing gear,' which then connects to other subsystems. In addition, attributes for the important nodes can be seen, such as the short branches leading away from the central nodes, whereas the longer branches indicate subsystems or a concatenation of requirements. The entire network structure is indicative of an architecture, and research regarding automatic or partial architecture generation based on such results is currently being conducted in follow-up research projects.

For the requirement structure, we see a different picture compared to the entity level. For one, the structure of the requirements does not show a clear order in the form of a tree or a star. This is due to the fact that its origin is the underlying connections, and as such, more edges exist for most nodes. Also, the crowded core of the requirements structure that expands outwards in a few directions indicates that the integrated center branches out into a few requirement subcategories. In addition, the requirement structure shows more loops and circular references than the entity level. This can be potentially problematic since circular connections can connect one requirement to itself through multiple other nodes, which can make changes and or verification more difficult and potentially re-work-inducing. Thus, such loops are starting points for analysis to evaluate their dynamics (also see Load metric in Chapter 7 and its application in Chapter 10).

8.4.2 Fragments and Potential Anomalies

Regarding fragments, the disconnected parts on the perimeter of the entity/NLP structure, while correctly identified and included, are caused by term inconsistencies in some cases. Improvements regarding the uniformity of the used terms, as well as the inclusion of relations and potential synonyms, could address some fragments. Yet, there are also disconnected parts that refer to specific aspects on a semantic level. For instance, one fragment refers to 'clearance requirement,' and since this specific compound term is a semantic descriptor for another statement, no connection can be made. Such edge cases require further analysis but do not significantly inhibit the function or validity (see 8.5) of the approach, but instead are improvement and expansion possibilities.

Furthermore, related terms that share a core or word as part of a compound have to be evaluated regarding their connection possibilities. For example, terms such as 'landing gear' and 'landing gear system' could potentially be considered the same term and thus their nodes combined, but this depends on semantics and background information. While such combinations are possible, they cannot be generalized, and with a growing network/structure size, the chance that a term appears in two different subsystems, despite not being the same component or entity, increases as well. Such cases have to be carefully considered and potentially need to be decided on a case-by-case basis or dependent on the system at hand.

The requirement structure does share the fragments with the entity structure since the former is a direct result of the latter, and thus, the edges that are not connected in the requirement structure correspond to the fragments within the entities. Yet, since the requirements appear as smaller fragments or just single, disconnected nodes, their analysis possibilities are different compared to the entities. For example, the evaluation and overall structure of a requirement network can be an indicator of the underlying document and inform about the cohesiveness of the specification as a whole. Thus, the level of the requirement structure can be used to analyze not only the potential architecture but also the document that it results from. This possibility, in connection with subjectivity, for example (see Chapter 10 for quantification), can be integrated into the analysis process as well to assess not only the complexity of the content but also the presentation and organization.

8.4.3 Metric and Numerical Implications

The metrics applied to the results (Figures 8.5., 8.6, and 8.7) also provide additional insights into the different levels of the created data. It has to be noted here that the third case study provides a more detailed metric application, which is why this subsection is kept shorter.

The density shown for the requirement structure was used to assess the overall population of the network. First, the number of introduced edges does not increase at the same rate as the number of possible edges, which follows an exponential function, despite both having a steady decline in density on average. Thus, plotting density together with the potential minimum density shows that, at first, the two metrics are closer together, and with increasing progress, the density of the network deviates from the minimum. Since the minimum density can be used as an indicator of a simpler topological structure, it is used as a baseline for comparison. This means that a deviation from this minimum indicates a higher topological complexity. Thus, the progress we see in the graph describes a structure that is less simple than it could be. Lastly, the graph also shows that while an overall trend exists, singular outliers have a disproportionate impact on the density by either greatly reducing or increasing it. These above-average impact requirements could point to nodes that have a significantly higher or lower degree of connectivity. Such outliers might be worth evaluating since high connectivity could create problems for verification, for example, whereas low connectivity or no connections at all could suggest erroneous or faulty requirements.

Figures 8.6 and 8.7, the spectral entropy and Cyclomatic Complexity metrics, calculated based on the entity network, show constant increases. While this is to be expected when a network such as the one at hand is expanded, the slope or partial slopes are indicators for overall behavior and changes during the growth of a network. Both metrics show similar small amplitude changes where some requirements add significantly more complexity/entropy than others. For the Graph Energy, this indicates nodes that increase topological complexity, whereas for the Cyclomatic Complexity, it can be either an indicator for a disproportionately high amount of edges added or the creation/addition of parts. Thus, these anomalies and amplitude changes potentially merit further investigation as to their nature and potential use.

The discussion and insights above conclude the results of the first case study. The outcome is also included in the verification and validation chapter (Chapter 11), where it is applied to confirm or reject the hypotheses of this dissertation. The last step in this chapter is to provide verification of the algorithm and approach application to assess and prove that the generated results are correct and, if not, how much improvement potential exists.

8.5 Verification of the Results, Limitations, and Conclusion

In order to verify the correctness of the results presented in this chapter and concurrently verify the approach that was developed in Chapter 6 and applied here, a set of metrics and criteria was defined to determine the quality of the results. As a guide for this verification, the accuracy and quality measures summarized by Derczynski (2016), as well as Nakache, Metais, & Timsit (2005), were used. Thus, Precision and Recall (PnR) are applied as a metric for correctness. This approach assesses algorithms and applications that target finding a set of items in a specific corpus. This objective is in line with the entity elicitation of the NLP approach of this dissertation, and thus, PnR was chosen. In order to conduct this assessment, a blind evaluation was conducted. In this evaluation, a human subject was presented with the requirement specification of the Skyzer UAV and tasked to apply the rules and entity elicitation approach by hand. The results of the human subject were then compared to the NLP output to calculate the precision according to the following formula:

$$P = \frac{|true\ positives|}{|true\ positives| + |false\ positives|} \tag{8.1}$$

The precision was calculated for the entity elicitation as well as structure definition on the NLP/entity level. This way, not only could the quality of the NLP results be checked, but also the structure definition based on the rules set in Chapter 6 for the implemented approach. The result of the precision assessment was that the approach correctly identified 387 out of 388 entities, yielding a precision of 99.74 percent for the elicitation of the entities. For the structural definition, the results were slightly lower but still high, with 383 out of 388 entities correctly linked/structurally assigned. These numbers yield a precision of 98.71 percent for the elicited structure. Since the requirement structure is a direct result of these two precision values, its precision and thus correctness is in line with these numbers.

For the recall (Equation 8.2), it is important to keep in mind that due to the imposed matching rules and inherent rigidity/structure of requirements, the NLP algorithm does not identify false negatives as it finds all subjects/objects, and thus only tends to identify something as an entity that is incorrect or part of another compound. As such, the recall score of the approach was 100 percent or 1.0 for both the entity and structural levels. Yet, due to the mentioned limitations, and as noted by Derczynski (2016), the recall score in such cases adds little value.

$$R = \frac{|true\ positives|}{|true\ positives| + |false\ negatives|} \tag{8.2}$$

To balance the two metrics, it is common to combine PnR with a weighted harmonic mean in the form of an F-Score (Van Rijsbergen, 1974). To achieve this, a β value is used in Equation 8.3 as a weight factor. Using a weight of one, the F1-Score of the approach applied in the case study is 99.88 percent on the entity/NLP level and 99.35 percent on the structural level.

$$F_\beta = (1 + \beta^2)\frac{P \cdot R}{\beta^2 \cdot P + R} \tag{8.3}$$

While the numbers shown above are significant and high regarding the accuracy of the NLP approach, it is crucial to consider the fact that while the NLP and algorithm were accurate in conducting the analyses and elicitations, the results are only as good as the matching rules and patterns that are set in the beginning. As outlined in Chapter 6, while subordinate clauses and enumerations/lists can be identified and structured, it is possible that the actual network resulting from them does not match the built structure as defined by the rules. For instance, it is possible that a subject is followed by two parallel objects listed and connected with 'and,' which creates a fork in the network. Now, the continuation of the network after the fork, if the sentence or requirement continues, connects the following entities to the last node in the fork. In some cases, it is possible though that all nodes of the fork should be connected to the subsequent entities. Figure 8.8 depicts these two different scenarios. Such distinctions have to be addressed when they occur, and respective rules must be defined when necessary.

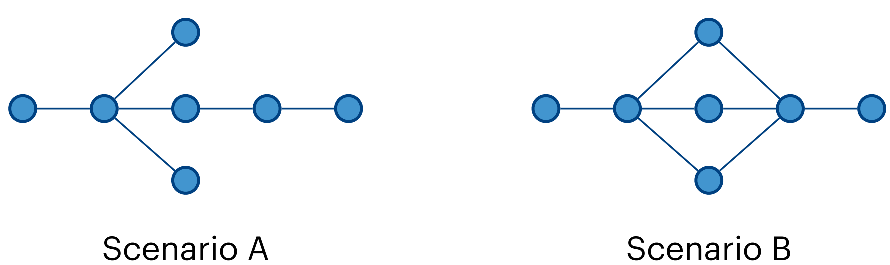

FIG. 8.8 - DIFFERENT STRUCTURAL INTERPRETATION POSSIBILITIES

In addition, while the precision score shows very accurate results of the algorithm, it does not mean that the identified entities, while correct, are useful in all cases. This limitation applies to all terms that do not appear in a dictionary or general database. For example, requirement 46 reads, 'The landing gear should be compatible with the Model 25 NSN 1730-ND-567-177GxW, CJ67D0250-1, and NSN 1730-00-854-2237RN.' In this requirement, all entities are identified, but since the standards and abbreviations are not words themselves, but rather abbreviations and letter combinations, they are not analyzed correctly. The resulting entities are 'landing gear,' 'Model,' 'NSN 1730-ND-567-177GxW,' 'CJ67D0250,' 'NSN,' and '00-854-2237RN.' While the elicited entities are correct, as per the rules, we clearly see that the NLP does not correctly separate or combine the standard definitions nor identify the root nouns within the terms. While this is a limitation of the approach, it is to be expected due to the foundation and NLP cores used. Furthermore, such errors could be easily eradicated by providing the standards and abbreviations to look for and identify. This way, the inclusion of a knowledge base or ontology could also be considered on a smaller scale.

Lastly, as already mentioned in 8.4.2, semantic interpretations are not yet part of the entity nor structural elicitation. This affects the mentioned 'clearance requirement,' for example, and also pronouns such as 'that' or 'which.' Since these pronouns replace an entity in another part of a sentence, the connections could be defined through dependency parsing, but the connections made with this approach have to be validated first and were not part of this research. Moreover, avoidance of pronouns is another possible option. Even involving a human to replace flagged pronouns with the respective term can be considered. Although replacements of pronouns might reduce the readability of the requirements, it would remove possible ambiguities and assist the function of the NLP approach.

All in all, this chapter showed the successful application of the developed NLP approach. Based on the scores as well as shown figures, the results can be considered correct and are used in the verification and validation chapter to address the hypotheses that pertain to the NLP aspects of this dissertation. To address the remaining hypotheses that are not exclusively NLP related, two other case studies were conducted that are described in the following chapters.

BIBLIOGRAPHY OF CHAPTER 8

Ballard, M., Baker, A., Peak, R., Cimtalay, S., Blackburn, M., & Mavris, D. (2020, 7-14 March 2020). *Facilitating the Transition to Model-Based Acquisition.* Paper presented at the 2020 IEEE Aerospace Conference.

Blackburn, M. (2018). *Transforming Systems Engineering through Model-Centric Engineering* (SERC-2018-TR-103). Retrieved from https://apps.dtic.mil/sti/pdfs/AD1048002.pdf

Derczynski, L. (2016). *Complementarity, F-score, and NLP Evaluation.* Paper presented at the Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16).

Jupyter. (2023). Retrieved from https://jupyter.org

McCabe, T. J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering, SE-2*(4), 308-320. doi:10.1109/TSE.1976.233837

Nakache, D., Metais, E., & Timsit, J. F. (2005). *Evaluation and NLP.* Paper presented at the International Conference on Database and Expert Systems Applications.

NetworkX. (2022). Retrieved from https://networkx.org

pandas. (2022). Retrieved from https://pandas.pydata.org

pyvis. (2022). Retrieved from https://github.com/WestHealth/pyvis

Sinha, K., & de Weck, O. (2013). *A network-based structural complexity metric for engineered complex systems.* Paper presented at the 2013 IEEE International Systems Conference (SysCon).

Van Rijsbergen, C. J. (1974). FOUNDATION OF EVALUATION. *Journal of Documentation, 30*(4), 365-373. doi:10.1108/eb026584

CHAPTER 9: CASE STUDY 2 - TOPCODER CHALLENGES

"If you want to have good ideas, you must have many ideas.
Most of them will be wrong, and what you have to learn is which ones to throw away."
Linus Paulin

The second case study was planned to bring the approach development together with the complexity-focused Hypotheses 4 through 7 and the content of Chapter 7. To archive this, a crowd-sourced dataset was selected that provided a sufficiently large foundation to yield valid insights. By applying the NLP approach as well as complexity assessment metrics, the correlation of specific aspects was planned to be evaluated. Unfortunately, the chosen dataset turned out to be unsuitable for the described plans, and as a result, different insights were possible. Despite the lack of success with this case study, the results and insights gained shall be described in this chapter since the lessons learned are still valuable and potentially crucial for future extensions of the research presented in this dissertation. To showcase the content of the case study, application, and results, this chapter is divided into four sections. The first section (9.1) describes the case study content and circumstances in detail. Based on these conditions, the second section (9.2) describes how the approach was applied before section 9.3 outlines the results of the application and the obstacles encountered. The insights from these three sections are then used to draw a conclusion, including the discovered limitations, in the last section (9.4).

9.1 Case Study Content and Circumstances

The dataset chosen for this case study was the result of a crowd-sourced coding challenge on the Topcoder website ("Topcoder," 2022). Through this website, challenges can be publicly submitted in order to receive solutions from participants for specific coding and data science challenges. From the website, a total of 4,908 challenges were retrieved in late 2021. The challenges were divided into projects, which allowed for analysis on a project and task level. Overall, 45 projects were retrieved. For each task/challenge, a plethora of data was retrieved as well, ranging from data points regarding the challenge in general all the way to the libraries, platforms, and technologies included in the individual listings. The result was over 600,000 data points that could be used for analysis. To illustrate the breakdown of the dataset, including projects and tasks, Figure 9.1 shows the structure of the data. The figure shows how the dataset is divided into projects, which in turn are divided into tasks. The number of tasks per project varied and is discussed below.

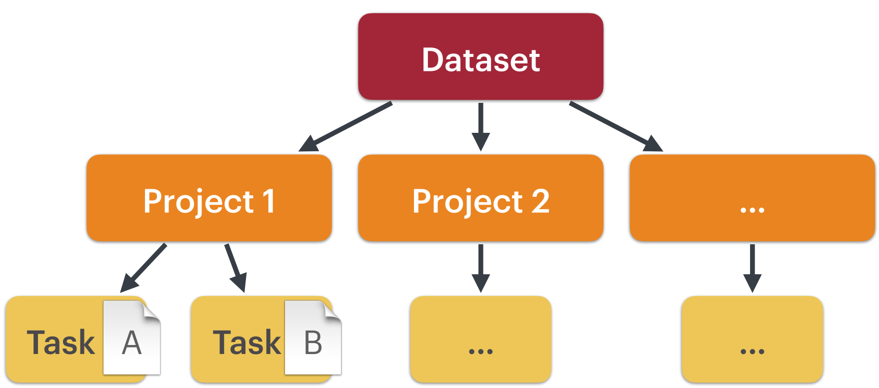

Fig. 9.1 - Dataset Structure including Projects and Tasks

Each task included specific instructions regarding the requirements of the challenge, technologies to use, documentation guidelines, submission guidelines, and testing approaches. Not all of these aspects were defined for every challenge, but the categories were available nevertheless. The content of the instruction files was available in plain text form with one file per challenge, no matter the status, submissions, or outcome. Also, in addition to the task descriptions, the project descriptions were included where available and necessary, bringing together the two levels (project and task) to increase comprehension and reduce ambiguities. Thus, the dataset was used in conjunction with the individual files as per the process and application in the next section.

9.2 Approach Application and Process

With the aforementioned data at hand, a plan was set for the case study to scientifically conduct tests that would allow for a conclusion to address the remaining hypotheses. To do this, in a first step, useful information was selected from the dataset since not all data points were useful. Thinking back to the connection and purpose that underlines the research in this dissertation, two variables in the dataset were of particular importance: the number of submissions for each challenge and the success rate of the submissions. These two aspects can be directly tied to the structural impact that the challenge requirements have on the outcome, which could then possibly be scaled to larger projects in a subsequent step. Therefore, the following logical flow was defined: 1) divide the dataset in two categories, one with submissions and one without; 2) process both datasets with the NLP algorithm and the complexity metrics; 3) analyze the correlations for the complete dataset between the metrics and the status if a submission was submitted or not; 4) analyze the correlations for the submission-only dataset between the metrics and the status if a submission was submitted or not; 5) analyze the correlations for the submission-only dataset between the metrics and the status if a submission was successful; 6) analyze the correlations for the submission-only dataset between the metrics and the status if a given project was more than 50 percent successful or not. Figure 9.2 visualizes this analysis.

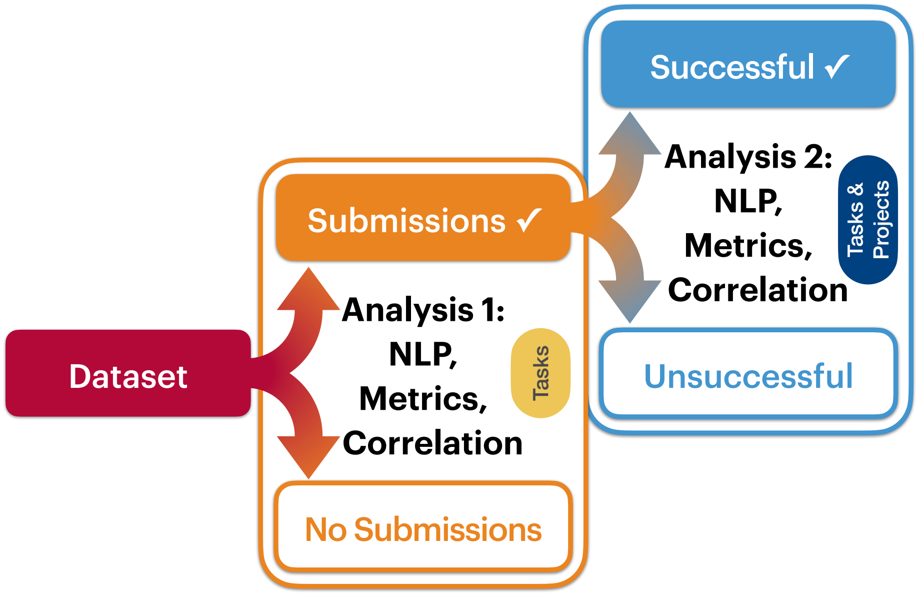

FIG. 9.2 - CASE STUDY ANALYSIS PROCESS STEPS

As the process shows, the two analysis levels would enable and yield three potential correlations: the first correlation on the task level in combination with the submission status, the second correlation on the task level in combination with the success status, and the last correlation on the project level in combination with the success status. Herein, the success of the projects was based on the average task success within, meaning that projects with more than 50 percent successful tasks were considered successful and vice versa.

It has to be noted here that for the task descriptions, continuous text paragraphs were split into individual sentences that were considered requirements. This had to be done to allow for the second layer of Figure 6.7 to be created since, without individual statements, no network could have been generated on the requirement level, only on the entity/NLP level. This does not apply to the project descriptions since no individual requirements were included here, and thus, only the entity/NLP layer was used for the analyses.

With the outlined plan, the application of the NLP approach and metrics was conducted. For this, the individual tasks and projects were split into task descriptions and project descriptions, which were processed by the algorithm separately. Like in the first case study, no knowledge base was used due to the fact that, for one, none was available, and two, the context and circumstances of the tasks/projects differed significantly, which would not allow for the application of a universal knowledge base or ontology. The results of the NLP algorithm were then used to apply the metrics, similar to the first study. With the results, correlation analyses were planned to be conducted. Yet, after running the algorithm and approach to this point, key insights emerged that contradicted and invalidated correlation analyses. These insights are described in the next section.

9.3 Results and Impediment Description

The results received from running the NLP algorithm on the different task and project descriptions yielded the expected results and produced the networks that we also saw in the first case study: for the tasks, entity and requirement networks were generated, and for the projects, only entity networks were produced. Also, initial precision and recall analyses of the networks showed similar numbers to the first case study, with slightly reduced (5 percent on average) precision levels on the structural level for task challenges that described requirements in continuous text instead of bullets or lists.

Unfortunately, after the generation of networks, during the analysis of metric results, two major limitations emerged that made interpretation and insight generation invalid.

First, the size difference based on word count between tasks and especially between projects led to the fact that the results of the metrics were scattered over a wide range of values. This is due to the fact that descriptions that were more detailed yielded more nodes and edges, which did not allow for comparison with smaller networks, for example. Such fluctuations could have been accounted for by using a calibration factor, as also shown by Salado & Nilchiani (2014), as well as Sinha & de Weck (2013), for example. Yet, in order to define a calibration factor to use with the given dataset, an empirical definition would have been required first, which was unavailable due to the novelty of the dataset.

Second, the subjectivity stemming from the person or team writing the descriptions was another factor that influenced the results. For instance, some people or teams described their requirements in line with the standards, while others wrote continuous text that sometimes included cross-references. This subjectivity and additional variation added to the limitations of cross-project comparison and sometimes even between tasks.

Lastly, the number of submissions received for each task, no matter if successful or not, varied greatly within a range from 0 to 1,637, with an average of 3. This adds an additional limitation since a higher number of submissions increases the likelihood of a successful one being among them. One could argue that despite a higher number of submissions if a correlation between a metric and the results existed, it would still emerge. While this argument is correct, the limitation created by the probability factor, while not as severe as the first two limitations, still undermines the validity of any result produced and makes it difficult to defend.

These factors and their impact meant that for the networks/tasks that could be correctly generated, the sample size for each group that could be compared and assessed for correlation would have been too small to be valid. Cross-project comparisons were deemed entirely impossible. As a result, the insights and conclusions outlined in the next section were drawn, and an alternative case study was chosen. It has to be noted that while this case study turned out to be unsuccessful and not valid at this time, revisiting it in the future might be possible if the limitations can be resolved or considered with aforementioned solutions, such as a calibration factor, for example.

9.4 Conclusion and Implications

While the inability to produce valid results in this case study is not ideal, a few important insights and conclusions can still be drawn from the process. First, the subjectivity influences on the NLP algorithm are a factor that has to be accounted for. Second, the size and comparison basis are aspects that needs to be considered when pairwise comparison or group analyses are concerned. Third, the lack of uniformity and resulting impossibility of conducting cross-project and or task comparisons is a factor that can occur under other circumstances as well. To elaborate on these three factors, this section addresses them in the following paragraphs before concluding the second case study.

As shown above, the influence of subjectivity and individual writing can have a major impact on the quality of the results as well as their validity. Thus, it has to be stressed that requirement statements are recommended to follow a pattern and are supposed to be written carefully to prevent errors later down the line. Taking into account the effect of subjectivity further reinforces the importance of these standards and rules. Also, adhering to a specific standard or at least carefully phrasing requirements in a structured and clear way can also affect the two points below. For instance, if the standard that a certain specification is written in accordance with is shared with another project or specification, a mutual foundation is provided, and fewer variable factors can affect the comparison as a result. Hence, this case study has shown that while the NLP approach functions, it is not immune to circumstantial and variable factors, which need to be considered.

In addition, the size and compatibility of projects and specifications, in general, has to be considered and potentially accounted for, if possible. Factors for calibration can be one way to address this consideration, but if not feasible, careful comparison and selection of a significantly large mutual foundation is necessary. Breaking up the system into subsystems and or smaller units that can be handled independently while retaining the necessary interfaces is another option that can be chosen if the modularity of the system permits. By separating functional units, the different modules can be compared regarding their architecture, structure, and or complexity while mitigating the effect of the differences that might exist on a system-wide and or assembly level.

Lastly, the uniformity aspect pertains to the quality of requirements and affects the compatibility significantly. While it is hard to generalize the effects, adhering to standards and recommendations as outlined in the previous chapters is a secure way to create a sound foundation for comparison. If such a uniform approach is not possible due to the origins of the different datasets, careful consideration and potential pre-processing are necessary to mitigate the effects if a comparison has possible merit.

All in all, this case study shows that as far as the NLP aspects of the approach and algorithm are concerned, a case dependency exists that needs to be factored in when multiple projects, specifications, and or systems are considered and compared. This problem is not as grave and risky when a single system is assessed at multiple stages in the systems development process since the comparison basis is provided and uniform. Therefore, this case study can be concluded with the insight that the approach is valid and fully functional, but scaling, transfer, and comparisons are not simultaneously given and have to be carefully assessed regarding their circumstances, environmental factors, and influences.

Bibliography of Chapter 9

Salado, A., & Nilchiani, R. (2014). The Concept of Problem Complexity. *Procedia Computer Science, 28*, 539-546. doi:https://doi.org/10.1016/j.procs.2014.03.066

Sinha, K., & Weck, O. L. d. (2013, 15-18 April 2013). *A network-based structural complexity metric for engineered complex systems.* Paper presented at the 2013 IEEE International Systems Conference (SysCon).

Topcoder. (2022). Retrieved from https://www.topcoder.com/

CHAPTER 10: CASE STUDY 3 - MOLECULAR INTEGRATION EXPERIMENT

"The man who is certain he is right is almost sure to be wrong,
and he has the additional misfortune of inevitably remaining so."
Michael Faraday

With the insights of the first and second case study, a third and last experiment was chosen to address the remaining hypotheses without the limitations and validity issues that emerged in Case Study 2. To achieve this, a controlled experiment and its results were chosen that allowed for a logical and analogous interpretation to address the remaining aspects. The experiment was initially conducted by Pugliese (2018) to measure the difficulties encountered by human subjects when dealing with different levels of complex systems and the integration thereof.

To structure the case study as well as the results and discussion, the following chapter has been divided into four sections excluding this introduction, which also addresses the analogies used to justify the use of the experiment as a case study. The first section (10.1) addresses the setup of the experiment and how it was conducted. Following this description, the second section (10.2) provides all the results in numerical as well as visual form. These results are then discussed and interpreted in the third section (10.3) before being concluded by the fourth and last section (10.4).

The overarching goal of the experiment initially was the evaluation of the system integration effort represented by time. In the experiment, the integration of a system was represented by an assembly task containing molecules. This assembly allowed for the measurement of complexity with spectral structural metrics, which were then assessed in conjunction with the recorded times. During the experiment, subjects were asked to manipulate objects representing hydro-carbon molecules in order to reproduce an assembly of the given objects in a tridimensional environment. To enable the tests, the assembly was provided for reference to each subject in a similar environment through the software application Blender ("Blender," 2022).

In order to use the described experiment for the research in this dissertation, the frame and content of the experiment had to be assessed for suitability. Since the experiment by itself did not concern RE or any of the adjacent tasks and topics, such as change management, for example, an analogy had to be defined to make the experiment useful and valid. Of course, such an analogy introduces limitations, which are described in 10.3 in detail.

As for said analogy, the integration task and the structure of the molecules themselves were found to be similar to aspects of the graphs and networks that are generated by the NLP algorithm of the developed approach. In addition, since the experiment by Pugliese (2018) dealt not only with the individual molecules but also with their integration, two layers of potential interpretation are enabled. As for the analogous connections, the different molecules were examined, and it was discovered that the structure of the included molecules (and we argue that this applies to chemical molecules in general), once translated into a two-dimensional structure, is similar to parts of the networks generated and seen in Chapter 8, for instance. To illustrate these similarities and the resulting analogy, Figures 10.1 through 10.3 show exemplary congruencies that illustrate the analogy. In the figures the layers and dimensions depicted in Figure 6.7 are considered in a top-down order.

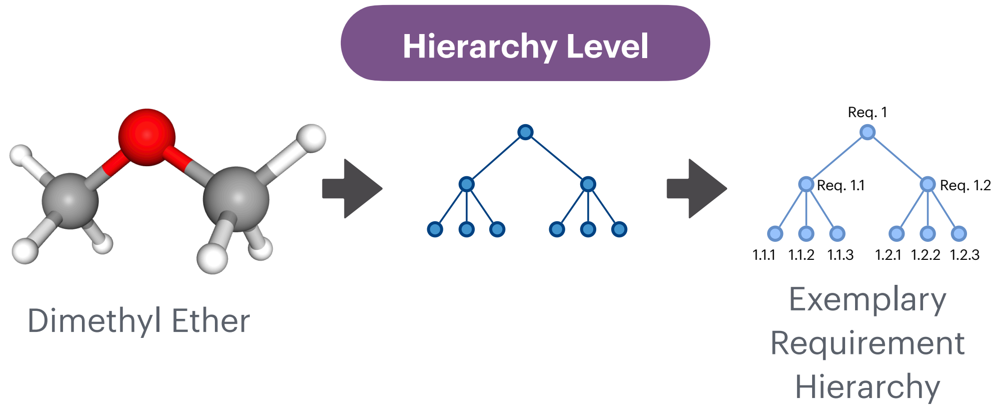

FIG. 10.1 - MOLECULE AND REQUIREMENT STRUCTURE SIMILARITY (HIERARCHY LEVEL)

As Figure 10.1 shows, the molecule shows an analogous structure to requirement hierarchy breakdowns when transferred into a graph representation. While the case studies conducted as part of this dissertation did not include hierarchies due to the limitations described, this analogy is still important to consider since a hierarchy is one of the layers, and the results from this case study can be considered analogous for the hierarchy level as well.

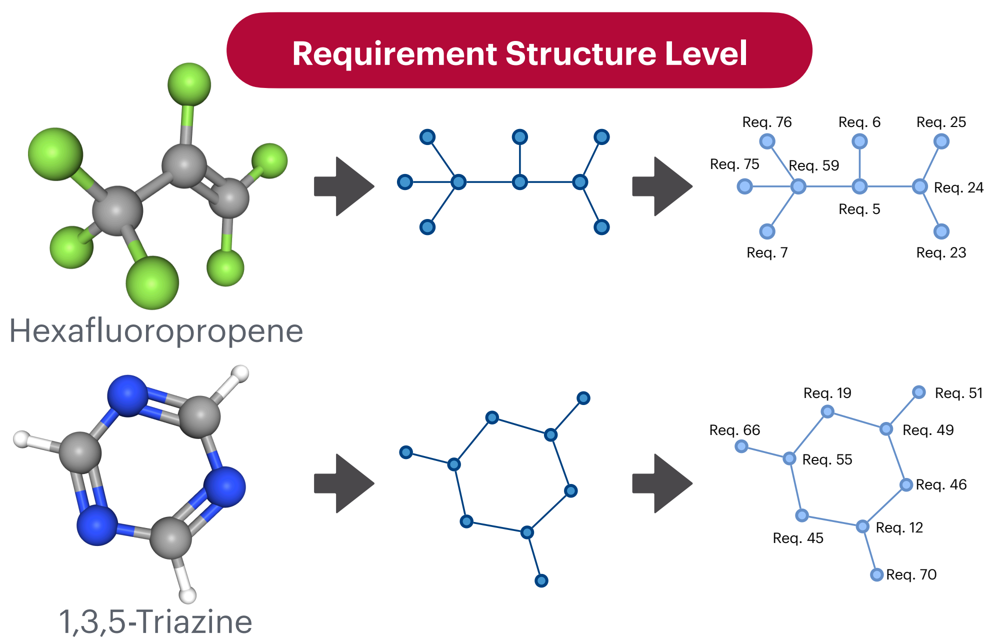

FIG. 10.2 - MOLECULE AND REQUIREMENT STRUCTURE SIMILARITY (REQUIREMENT LEVEL)

On the level of the requirements, two examples from the second case study were used to show the analogy between the molecules and the graph structures elicited from a requirement specification. The top part of Figure 10.2 shows a subset of the branches in Figure 8.4, which includes the depicted requirements. The bottom part of the figure visualizes the analogy with nine requirements that are connected in the dense center of Figure 8.4. Naturally, the structures within the requirements network are not disconnected and have further connections on some or all of the nodes. However, these additional connections do not invalidate the analogy or the case study since, as described, not only the metrics for the individual molecules but also for the integration tasks were considered and can thus be analyzed.

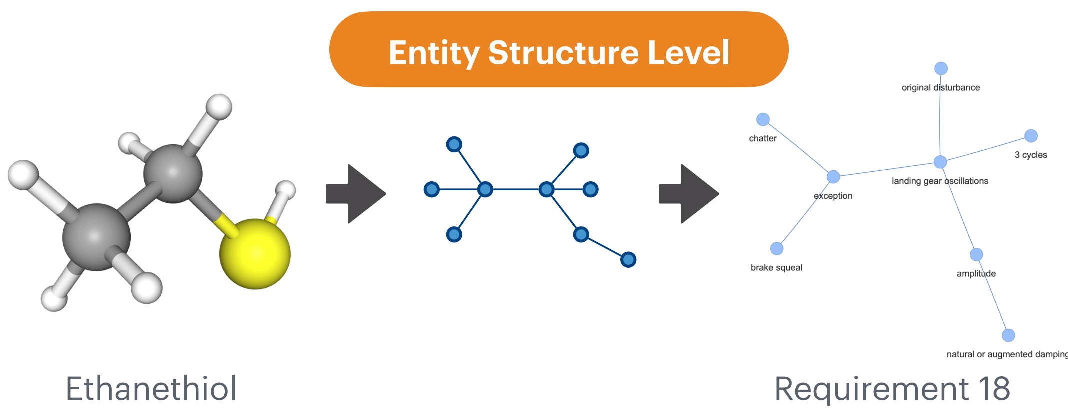

FIG. 10.3 - MOLECULE AND REQUIREMENT STRUCTURE SIMILARITY (ENTITY/NLP LEVEL)

On the bottom layer, the entity/NLP layer, the best example for the analogy is requirement 18 of the Skyzer UAV, which formed a fragment in the network. This fragment shows an almost perfectly (with the exception of one node) identical structure to the molecule Ethanethiol (also known as ethyl mercaptan), as shown in Figure 10.3. Hence, on the entity level, the analogy with the molecules holds true as well and the analyses of this case study can be seen as transferable (with the limitations discussed in 10.4).

Lastly, it shall be noted that similar studies have been conducted with successful interpretations by other researchers as well. One notable example is Sinha & de Weck's use of ball and stick molecule models to validate their structural complexity metric (Sinha & de Weck, 2016) as well as Alkan's application (2019). The reasons listed by these research groups for their analogous application are similar to the ones that apply here, namely the absence of large existing requirement sample sets, the immeasurability of complexity, and the cost/effort analogy described in Chapter 7. The validity and good results of these previous studies are further confirmation of the auspiciousness of the use of this case study.

10.1 Experiment Content and Setup

To outline the exact frame and conditions of the case study, an overview shall be given in line with the descriptions by Pugliese (2018). In the beginning, 23 subjects were given a practice task, including a tutorial for the integration that they were presented with in the actual study. This training was considered a learning task and was not timed. The untimed round also allowed the subjects to practice the integration and understand the environment before completing the timed tasks, thus preventing learning effects throughout the actual experiment.

Then, for the timed part of the experiment, the subjects were presented with ten tasks (A through J) in a randomized order. The ten tasks were divided into two groups of high and low complexity. The two groups were also randomized separately to avoid that two subjects complete the tasks in the same order and to exclude that a single task was affected disproportionately by routine effects over the course of the experiment. Thus, a routine effect was spread out and did not interfere with the measurements as the growing complexity was opposing any routine. These precautions, in combination with the learning and practice example, allowed for a low interference of learning and practice effects with the measures of interest, i.e., the time to complete the integration tasks.

For the individual tasks, the time was recorded for each subject, yielding 230 data points, 23 for each of the ten tasks. If mistakes were made during the tasks, the subjects had to correct the errors, which would add additional time to their measured data. The measured times are shown in Table 10.1.

TABLE 10.1 - MEASURES EXPERIMENT TIMES

	Task A	Task B	Task C	Task D	Task E	Task F	Task G	Task H	Task I	Task J
Subject 1	74	364	73	81	110	109	111	170	518	600
Subject 2	70	53	130	86	244	303	288	289	369	439
Subject 3	94	53	99	93	141	141	124	302	302	482
Subject 4	153	191	171	232	195	253	319	504	501	578
Subject 5	87	36	68	140	145	197	279	255	231	435
Subject 6	85	21	36	114	107	106	166	206	444	471
Subject 7	43	56	64	52	112	152	219	245	149	207
Subject 8	65	40	90	83	136	152	202	399	340	276
Subject 9	102	25	153	46	141	71	240	153	176	397
Subject 10	37	20	48	43	106	118	141	163	369	277
Subject 11	113	69	172	162	133	193	174	317	449	481
Subject 12	53	97	102	171	81	131	117	147	206	189
Subject 13	35	52	72	116	130	422	90	284	155	455
Subject 14	56	53	120	157	139	174	188	396	296	344
Subject 15	103	38	73	66	162	250	170	250	335	233
Subject 16	48	105	56	139	121	132	129	328	285	460
Subject 17	72	40	121	124	194	179	153	218	353	337
Subject 18	83	36	92	69	128	270	129	276	236	357
Subject 19	76	62	131	66	126	146	246	333	352	344
Subject 20	67	130	104	132	167	142	135	279	283	418
Subject 21	54	31	88	149	127	149	147	324	353	407
Subject 22	85	30	109	47	150	214	218	291	292	211
Subject 23	78	39	60	66	135	152	244	273	312	378

In addition to the individual tasks, the statistical evaluation was calculated and summarized by Pugliese (2018), which is shown in Table 10.2. The p-value was assessed through a Kolmogorov-Smirnov one-sample test, comparing each task with a normal distribution to see if the selected samples fit the population. According to these values, all tasks except Task B are normally distributed given a significance level of 0.05 (Pugliese, 2018). Satisfying this condition shows that the number of subjects (in this case, 23) is large enough for it to be representative for the general population based on the normal distribution assumption.

TABLE 10.2 - STATISTICAL EVALUATION OF EXPERIMENT RESULTS (PUGLIESE, 2018)

Task	Mean	Median	Std. Deviation	p-value
A	75.35	74	27.05	0.9333
B	71.35	52	75.13	0.0284
C	97.04	92	37.44	0.7882
D	105.83	93	49.06	0.7580
E	140.43	135	34.75	0.3416
F	180.70	152	77.47	0.2158
G	183.87	170	62.84	0.8386
H	278.35	279	84.90	0.8663
I	317.65	312	100.45	0.8540
J	381.57	397	112.90	0.9863

All molecules and the foundation for the tridimensional blender models were obtained from the publicly accessible database of the National Library of Medicine, which can be found online ("PubChem," 2022).

From the PubChem database, the molecules were imported, interpreted as NetworkX data, and then represented in the Blender environment. In this interpretation, each molecular bond was treated as an undirected link. Also, double bonds were treated as single links between two atoms because all atoms were used without changes. Hence, no perceptible difference would have existed between different bond types. This circumstance is similar to the connections between entities and requirements in the first case study, as described in Chapter 8.

With the molecules available in the NetworkX format, the metrics and calculations could be applied. For this, based on the described layers, the molecules, as well as the integration, could be analyzed. For example, the molecule Ethanethiol in Figure 10.3, which has the PubChem identifier 6343, has the following structure once translated into its adjacency matrix:

$$A_{6343} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

For the integration level, the different molecule connections were used to create one integration matrix per task. In these tasks, each link between two molecules was identified by one cell in the matrix if a link was to be created. For instance, Task A had the connections (indicated by a black circle in the respective fields) shown in Table 10.3.

TABLE 10.3 - TASK A CONNECTIONS

	2187	1739	2986	4976	4204
2187	○	●	●	●	●
1739	●	○			
2986	●		○		
4976	●			○	
4204	●				○

With the foundation above, the metrics could be applied and then correlated to the times of the subjects. For this, the metrics listed below (as defined in Chapter 7) were applied to each molecule individually and then to the integration tasks as well (where possible). To calculate the integration values for the metrics (β_{ij}), the individual values (α_i & α_j) for the respective molecule pairs were calculated and then combined according to Equation 10.1:

$$\beta_{ij} = \sqrt{\alpha_i \cdot \alpha_j} \tag{10.1}$$

With this equation, the foundation to calculate all the metrics is given.

Thus, the following metrics were calculated and evaluated on the **molecular level**:

- Cyclomatic Complexity - total of all molecules in one task and average
- Graph Energy as entropy - average of the molecules
- Laplacian Graph Energy as entropy - average of the molecules
- Density - average of the molecules
- Absolute Density - average of the molecules

And the metrics below were assessed at the **integration level**:

- Graph Energy as entropy - calculated with equation 10.1
- Laplacian Graph Energy as entropy - calculated with equation 10.1
- Density - for the respective integration matrices
- Absolute Density - for the respective integration matrices
- Density Delta - deviation from the minimum density for the integration matrices
- Integration Load - defined as the loop count for each integration task

Lastly, it has to be noted that due to corrupt and unobtainable data caused by updates on the PubChem website, three molecules were not available anymore for computing. As a result, two tasks (C & H) could not be included anymore in the evaluation. However, the statistical evaluation described above still holds true and thus validates the data nevertheless.

With this foundation, all the metrics were calculated and visualized, which is described and depicted in the next section.

10.2 Results And Plots

In accordance with the two layers described for the applied metrics (see 10.3), the results below are divided into the molecule part (10.2.1) and the integration part (10.2.2).

10.2.1 Molecule Level Results

To organize this results section in accordance with the applied metrics, each metric below is plotted individually and briefly described. The full discussion, including the insights generated, can be found in Section 10.3. All metrics are represented as box and whisker plots. Since this case study set out to assess how the complexity of the molecules, as well as the integration tasks, affect the times of the subjects, the respective correlation factors are provided as well and also summarized in Table 10.4.

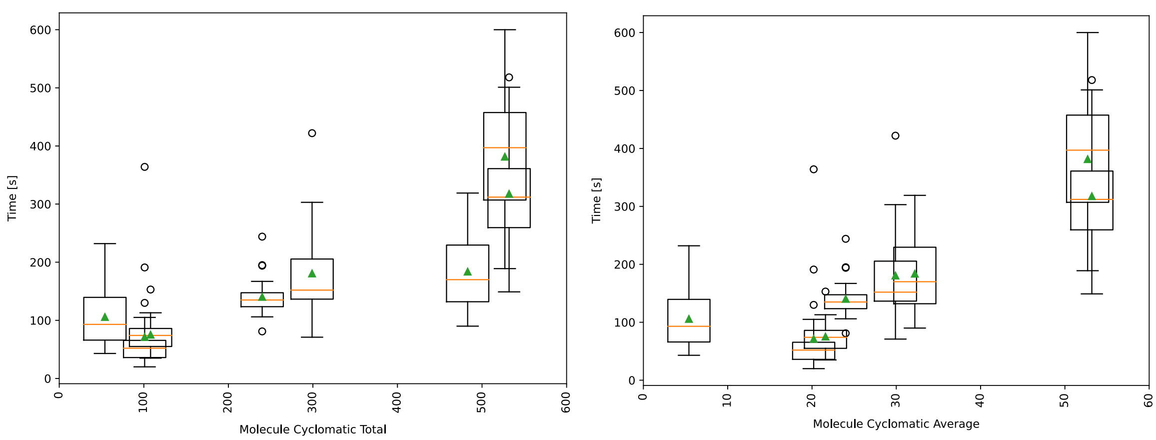

FIG. 10.4 - MOLECULE TOTAL AND AVERAGE CYCLOMATIC COMPLEXITY

As seen in Figure 10.4, the Cyclomatic Complexity levels on the left show an increase in the integration time with a rising complexity level. The Pearson correlation coefficient for the total Cyclomatic Complexity is 0.8919 and for the average 0.9125, respectively. The 95 percent confidence intervals for the given sample size are [0.504, 0.9804] and [0.5822, 0.9843]. These values indicate a moderate and most likely strong correlation for both.

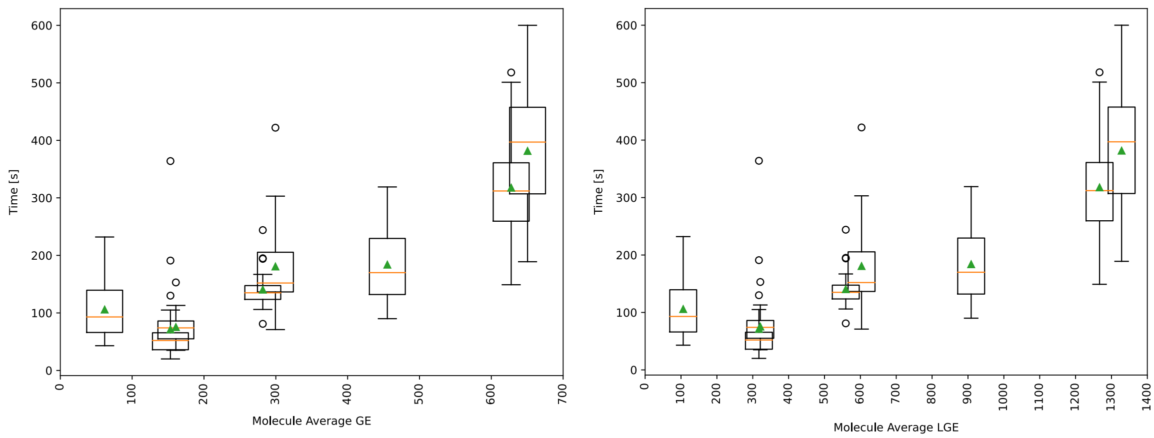

FIG. 10.5 - MOLECULE AVERAGE GRAPH ENERGY AND LAPLACIAN GRAPH ENERGY

As shown in Figure 10.4, the average Graph Energy (GE) and Laplacian Graph Energy (LGE) both show an increase in the integration time with a rising complexity level. The Pearson correlation coefficient for the average GE is 0.942 and for the average LGE 0.9426, respectively. The 95 percent confidence intervals for the given sample size are [0.7059, 0.9897] and [0.7086, 0.9898]. These values indicate a strong correlation for both metrics.

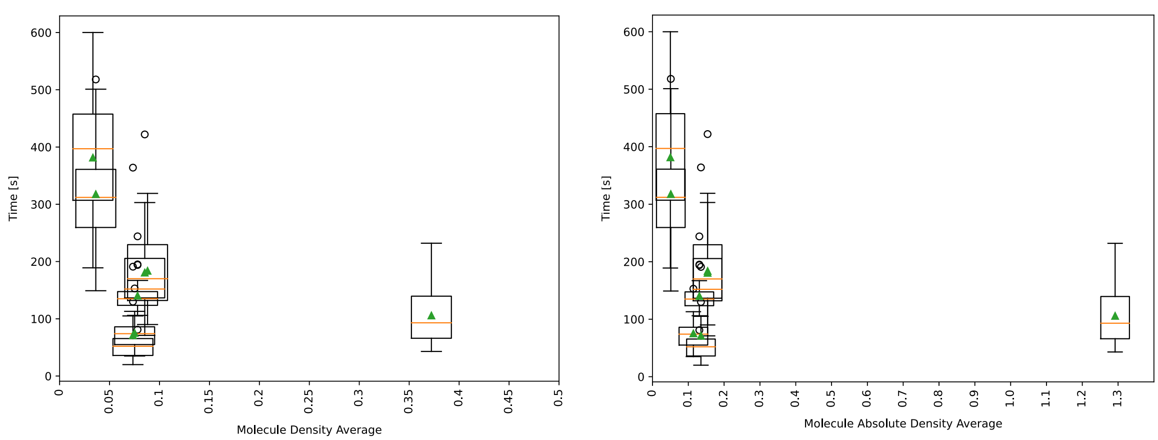

FIG. 10.6 - MOLECULE AVERAGE DENSITY AND AVERAGE ABSOLUTE DENSITY

As shown in Figure 10.6, the average density, as well as absolute density, show no clear trajectory for their correlation to the recorded times. This absence of a clear picture is also shown by the Pearson correlation coefficients, which are -0.4163 and -0.3446, respectively. The 95 percent confidence intervals for the sample size are [-0.8667, 0.4081] and [-0.8443, 0.4756]. As a result, these values indicate no significant correlation and are discussed in section 10.3.

10.2.2 Integration Level Results

On the integration level, the two tasks that were not usable anymore due to the missing molecules could be reinterpreted based on the recorded data for the GE and the LGE (see Pugliese, 2018) and are thus included in the two diagrams below only.

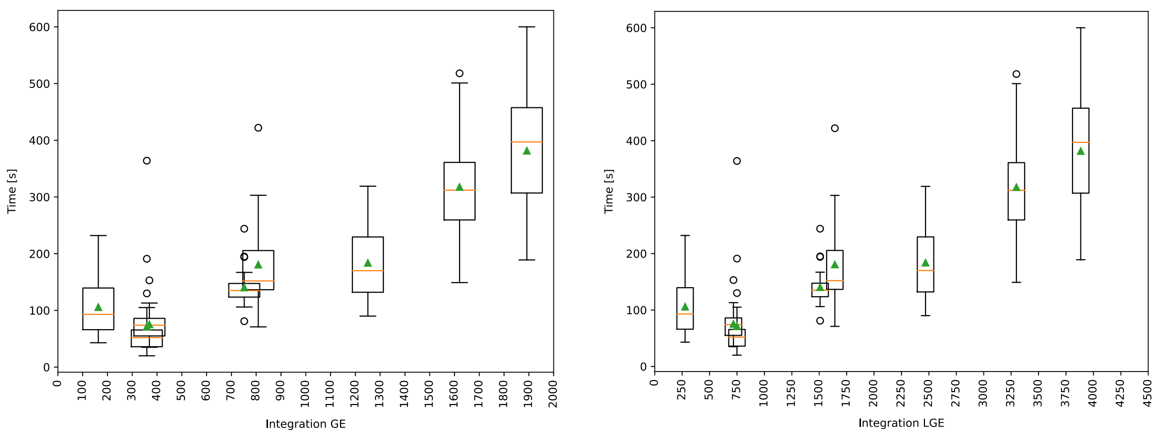

Fig. 10.7 - Integration GE and LGE

As shown in Figure 10.7, the integration GE and LGE both show an increase in the integration time with a rising level in complexity. The Pearson correlation coefficient for the average GE is 0.9545 and for the average LGE 0.9572 respectively. The 95 percent confidence intervals for the given sample size are [0.7631, 0.992] and [0.7758, 0.9925]. These values indicate an even stronger correlation for both than what was seen on the molecular level. In addition, looking back at the data produced by Pugliese (2018), the confidence interval for the GE shows an even higher lower bound with [0.7761, 0.9871] while being slightly lower on the upper end.

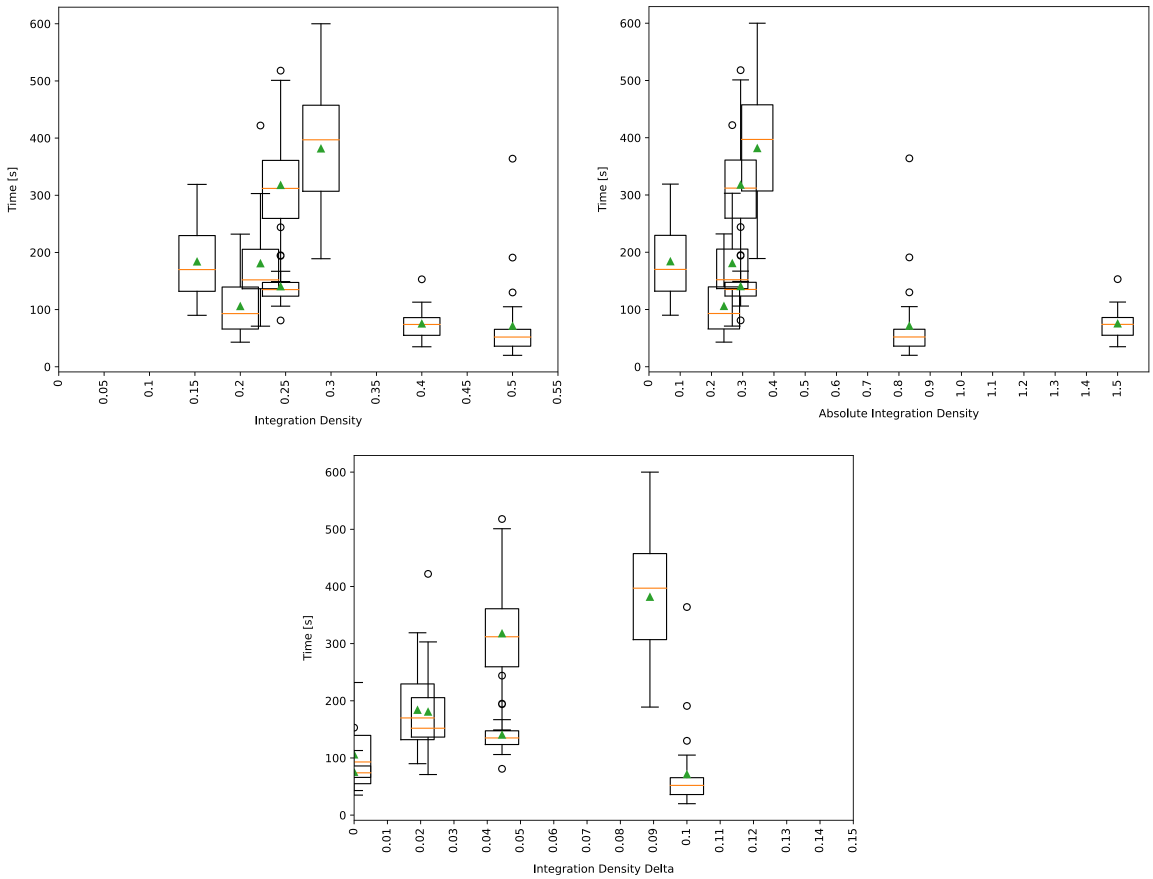

Fig. 10.8 -Integration Density, Absolute Density, and Density Delta

As shown in Figure 10.8, the integration density values reveal a picture similar to the density values on the molecular level. No significant correlation can be seen here either. The correlation factors are -0.3627, -0.4720, and 0.3626 with 95 percent confidence levels of [-0.8501, 0.4594], [-0.883, 0.3486], and [-0.4595, 0.8501]. To make sure, a quadratic regression was performed for the Integration Density Delta, since the shape of the curve looked potentially inverse quadratic. This regression yielded a coefficient of 0.6163, indicating a moderate correlation for an inverse quadratic function. This bears a resemblance to some of Pugliese's (2018) findings as well. As aforementioned, it is crucial to note that the integration level density is related to spectral density as outlined by Albert & Barabási (2002), for example.

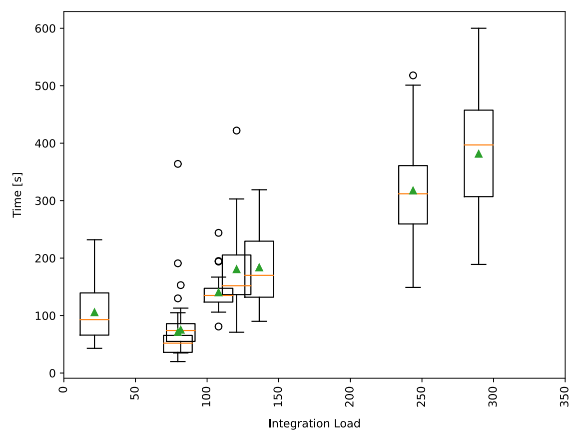

FIG. 10.9 - INTEGRATION LOAD

Lastly, the Integration Load shows a picture that is very similar to the two graph energy evaluations. The Pearson correlation factor here is 0.9546 with a 95 percent confidence range of [0.7636, 0.992]. This indicates a very strong correlation, even at the lower end of the interval.

To provide all results in a unified form, Tables 10.4 and 10.5 below show all the results, including their confidence intervals. The two sections for the molecule and the integration level are colored in accordance with the aforementioned list.

TABLE 10.4 - MOLECULE LEVEL RESULTS OVERVIEW

	Correlation Coefficient	95% Confidence Interval
Molecule Total Cyclomatic Complexity	0.8919	[0.504, 0.9804]
Molecule Average Cyclomatic Complexity	0.9125	[0.5822, 0.9843]
Molecule Average Graph Energy	0.9420	[0.7059, 0.9897]
Molecule Average Laplacian Graph Energy	0.9426	[0.7086, 0.9898]
Molecule Average Density	-0.4163	[-0.8667, 0.4081]
Molecule Average Absolute Density	-0.3446	[-0.8443, 0.4756]

TABLE 10.5 - INTEGRATION LEVEL RESULTS OVERVIEW

	Correlation Coefficient	95% Confidence Interval
Integration GE	0.9545	[0.7631, 0.992]
Integration LGE	0.9572	[0.7758, 0.9925]
Integration Density	-0.3627	[-0.8501, 0.4594]
Integration Absolute Density	-0.4720	[-0.883, 0.3486]
Integration Density Delta	0.3626	[-0.4595, 0.8501]
Integration Load	0.9546	[0.7636, 0.992]

Now, with the results complete, the actual analysis and interpretation can be addressed, which is described in the next section. Also, the presented results are used in the next chapter for the validation and verification of the hypotheses.

10.3 Interpretation and Discussion

To keep the organization of the results above, the interpretation and discussion are divided into three subsections. The first two subsections cover the results according to their appearance above, first the molecular level and then the integration level. The last subsection addresses limitations and current environmental boundaries of the results, including application.

10.3.1 Molecule Level Discussion and Insights

On the molecule level, as seen in Table 10.4, as well as Figures 10.4 through 10.6, the results showed moderate to strong correlations for the entropy-based and the Cyclomatic Complexity metrics. This is in line with the findings of Pugliese (2018) as well as Sinha & de Weck (2016), although the latter described an exponential function in their results. Yet, the results of this case study differ from Pugliese's since the assessment on a per-molecule basis was not considered in the original study (Pugliese, 2018). Thus, the implications thereof are discussed here.

First, the high correlations between the entropy and Cyclomatic Complexity approaches on a molecule basis suggests that not only the integration task by itself in isolation is connected to the effort and resulting time, but also the components within the task. This indicates that components and parts that are more complex can cause higher efforts even if the actual integration and assembly task is not as complex. Also, this effect can be an indicator that maybe additional features, such as symmetry, play a role since they affect the ease with which the entirety of a part or component can be understood, seen, and handled as a result. These findings are in line with what is discussed as perceived complexity (Grogan, 2021). This complexity is different from the inherent system complexity and affects the development effort in a different way since it also then includes factors such as personal aspects and organizational surroundings.

The absence of correlations regarding the density metrics is also an interesting aspect. For one, there seems to be no effect of the density value regarding the effort and resulting time. Yet, this absence does not necessarily have to be the full picture. Due to the fact that the number of possible connections grows exponentially with each node added, it is unlikely that the number of edges in a molecule grows sufficiently to not result in a decrease in density. Thus, a decline in density is to be expected with bigger networks. Yet, even the absolute density, which should not be affected by this limitation, as explained in Chapter 7, also shows the same absence of correlation. Hence, in order to further assess the connections and effects of density, further case studies with more controlled conditions that target such tests are necessary.

Lastly, it must be mentioned that the correlations found in this study are mostly linear, which contradicts other findings, such as Sinha's & de Weck's (2016), for instance. Yet, the small sample size of the data does not allow for a definite answer to confirm if the overall curve and correlation are entirely linear since exponential behavior is still possible with either more data points or a larger study in general, which could consider a wider range.

10.3.2 Integration Level Discussion and Insights

On the integration level, which has to be considered dependent on the molecule level for the GE and LGE (not for the other metrics), due to Equation 10.1, we see an even stronger correlation for the entropy-based metrics. This is to be expected due to the dependency, but the strong correlation confirms that the integration task and the molecule complexities do, in fact, behave similarly and thus have a similar effect on the effort and development. Thus, the same observations noted in the previous subsection for the molecular level apply here in an even more pronounced and more significant way, as shown in Figure 10.5.

For the network densities, the results are similar as well, with no significant correlations visible in the data. The absence of correlations makes sense since the integration tasks, while not affected as much by network size, do share the same characteristics of node/edge increase in numbers and thus are subject to the same limitations and circumstances. Yet, the density delta, which only was viable to be calculated on the integration level where the analogy to requirements being set and built holds, shows no correlation either. However, the slight quadratic correlation could be a confirmation of what was seen in the first case study, where first drastic changes in the density occur, which keep it close to the minimum (effectively lowering the delta), but after a while, the delta grows again with the requirement set, effectively increasing the effort and time that was and has to be invested. Yet, in order to fully confirm this tendency, a larger case study conducted just for these types of dynamics and hypotheses has to be conducted.

Overall, the results of the case study show clear pictures in some areas and also point to necessary extensions and expansions in others. In addition, the gained insights can be used to address the hypotheses in the next chapter.

10.3.3 Limitations and Peculiarities

Now, since the case study presented in this chapter did not work with requirements nor did it directly address RE, the limitations and restrictions of the analogy mentioned in the beginning have to be outlined. Furthermore, one peculiarity emerged and was visible in the dataset that shall be mentioned here since it might merit further investigation as well.

The biggest limitation factor of the conducted case study is the fact that no textual requirement specifications were involved. This means that the results obtained and the insights generated are only applicable to the RE space and the overall approach developed as part of this dissertation through transfer. While the transfer of these insights based on the analogies shown at the beginning of this chapter is valid since the structures contained in the requirement sets resemble the molecules, no direct interpretation is possible as to how complexity might affect RE, for instance. This also means that no specific task in the RE space or processes can be linked to the results of the presented case study. However, due to the general implications, we argue that higher complexity and or metrics, as measured, will cause higher effort and, as a result, require more time and consequently money, which directly applies to RE. On the other hand, if time is not available, the error rate will increase due to a lack of correction opportunities.

The second limitation is that the subjects in the case study had to deal with tridimensional constructs instead of the two-dimensional networks that are produced by the NLP algorithm, for example. This limitation is of theoretical nature though, since the structures of requirements, entities, and hierarchies could easily be represented in a three-dimensional space. However, the benefit of such a representation compared to a two-dimensional one has to be evaluated first since the additional dimension adds another variable. Yet, this does not decrease the validity of the results nor of the analogy since the only implications of the missing dimension could be a less pronounced effect and resulting reduced effect of the correlations. It is even possible that a complex structure represented in a two dimensional network will cause more problems and have an even stronger effect due to overlapping edges, a more crowded appearance, and or less freedom regarding perspective. If this effect turns out to be correct, the three-dimensional environment of the experiment in this case study was decreasing the effects and dynamics that would be seen with a two-dimensional setup.

Lastly, an additional limitation has to be noted regarding the sample and subjects selected. Due to the circumstances, all subjects had a more or less pronounced engineering background due to their affiliation with a technical institute. This means that we can assume a higher degree of affinity to technical and logical problems, which could have affected the outcome of the experiment. While it is unlikely that these attributes of the sample generated entirely different results compared to a more heterogenous group, the actual magnitude and or amplitude of the results might have been affected, which is why no specific curves or equations were provided for the correlations, for example.

The aspect of heterogeneity and personal affinities brings us to the last aspect of this case study, a noteworthy peculiarity that emerged: for all of the highly correlated metrics, no matter if at the molecule or the integration level, trends specific to the individual subjects can be seen. These trends mean that certain subjects seemingly had an easier time dealing with the higher complexity or metrics, and their times did not increase with a slope as high as others. While this is to be expected as a part of the dataset, it is worth pointing out due to its possible implications. For illustration purposes, Figure 10.10 show Subjects 4, 7, and 19 for the Integration Load metric (showing all 23 subjects would not have allowed for good readability).

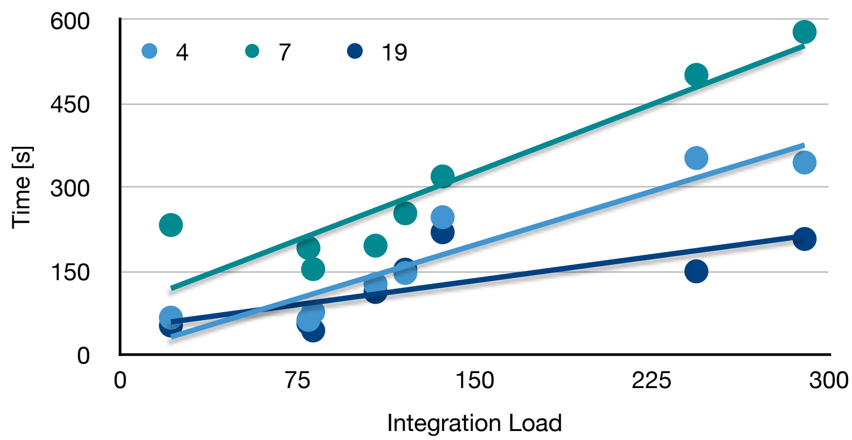

FIG. 10.10 - SUBJECTIVITY TRENDS FOR INTEGRATION LOAD EXAMPLE

As shown in the figure, there is a stark difference between the subjects regarding the slopes and inception points. Overall, Subject 19 seems to be affected very little by the increases in Integration Load, while Subjects 4 and 7 are affected almost equally as far as their increase is concerned, with Subject 7 having a much higher starting point. Such aspects and singularities can be indicators of personal affinities and, given the fact that the experiment was testing complexity and other metrics, possibly even general talent regarding certain tasks. While such aspects would have to be tested in specific and targeted case studies, they indicate that certain people might be dealing better with complexity and complex tasks than others. Such insights could be used, if confirmed in a valid way, to choose the right people for specific tasks. For instance, people with an affinity and consequently easier time working on complex tasks might be well suited to work with large and complex requirement sets in change management. On the other hand, having such people work on creating requirement specifications might not be practical since they might tend to underestimate the complexity and effort required, which is another aspect to consider when it comes to the concept of perceived complexity. Aspects of subjectivity, as well as perceived complexity regarding system design, have also been researched by Grogan (2021), who discusses the relation between performance, complexity, effort, and time.

10.4 Conclusion

All in all, the case study based on the experiment using analogous molecule integration tasks to test the effects of complexity on human effort was successful and yielded good results. For one, we have seen that the metrics of Cyclomatic Complexity, GE, LGE, and Integration Load correlate moderately to strongly with the effort that is required on the molecular as well as the integration level. This indicates that for RE tasks where humans have to work with such problems, the effort and resulting time necessary will be affected respectively in accordance with increasing complexity.

Density, on the other hand, has been shown not to have any correlation, which is assumed to be due to the fact that it depends on various factors that make comparisons difficult and, as shown in Chapter 7, might not be a suitable metric for the purpose of the presented research. Yet, having evaluated density in this case study confirmed the assumption and also added important details and expansions to the findings of the first case study.

While the way in which the case study was conducted introduces certain limitations, none of the discussed limitations are seen as crucial or severe enough to invalidate the analogy between the molecule and requirement structures. Hence, the results, findings, and interpretations presented in this chapter are valid, considering the given limitations. The limitations will also be considered for the validation and support/rejection of the hypotheses.

Furthermore, subjectivity and perceived complexity also emerged from the case study and are thus considered promising options for future work and research projects in line with existing research (Grogan, 2021). Now, with the results complete and all analyses conducted, the verification and validation chapter addresses the hypotheses set forth for this dissertation.

Bibliography of Chapter 10

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics, 74*(1), 47-97. doi:10.1103/RevModPhys.74.47

Alkan, B. (2019). An experimental investigation on the relationship between perceived assembly complexity and product design complexity. *International Journal on Interactive Design and Manufacturing (IJIDeM), 13*(3), 1145-1157. doi:10.1007/s12008-019-00556-9

Blender. (2022). Retrieved from https://www.blender.org

Grogan, P. T. (2021). Perception of complexity in engineering design. *Systems Engineering, 24*(4), 221-233. doi:10.1002/sys.21574

PubChem. (2022). Retrieved from https://pubchem.ncbi.nlm.nih.gov

Pugliese, A. (2018). *Development of Spectral Structural Complexity Metrics in Cyber-Physical Systems.* (Doctor of Philosophy). Stevens Institute of Technology, Hoboken, NJ.

Sinha, K., & de Weck, O. L. (2016). Empirical Validation of Structural Complexity Metric and Complexity Management for Engineering Systems. *Systems Engineering, 19*(3), 193-206. doi:10.1002/sys.21356

CHAPTER 11: VALIDATION

"The great tragedy of Science - the slaying of a beautiful hypothesis by an ugly fact."

Thomas Henry Huxley

In the previous chapters, we looked at the results and the insights produced in the case studies. Now, the hypotheses set forth in Section 5.3 can be addressed and validated. To conduct this validation, the first section (11.1) in this chapter outlines the approaches and criteria used. With this foundation, the second section (11.2) explains the support or rejection of the hypotheses, including applicable limitations. Lastly, Section 11.3 adds a discussion. Sections 11.1 and 11.2 are divided into subsections, one for each hypothesis.

11.1 Validation Approaches

11.1.1 Hypothesis 1

Requirement text can be categorized and structured based on contextual and or explicit content.

As shown above, Hypothesis 1 targeted the categorization and structure of requirements based on explicit and implicit or contextual content. This hypothesis was addressed in the first case study and is validated by the possibility and feasibility of generating a categorized approach that considers explicit as well as implicit information either separately or in combination was used as a criterion. This was to be validated through logical reasoning and human checks as per Table 5.2. It has to be noted that in this case (which also affects subsequent hypotheses), the term categorization was used not just as context categories, for example, but also on a syntactic or semantic level, such as entity (e.g., nouns and objects) elicitation for example.

11.1.2 HYPOTHESIS 2

Structure and or networks can be derived from categorized requirement texts and content.

The second hypothesis builds upon the first and, as a result, depends on it. Hypothesis 2 targets the generation of networks that are built on the categorized requirement texts and the content thereof. For the validation, the same feasibility approach was used as a criterion and checked by a human for validity, as per Table 5.2.

It is important to mention that the used open-source software and libraries were not assessed on a functional level but evaluated according to the quality of their results. This restriction also applies to all other validation approaches that include software or code that was initially generated by external parties.

11.1.3 HYPOTHESIS 3

A structure and frame for contextual interpretation and reasoning of requirements can be defined.

The third hypothesis addresses the contextual interpretation as a specific addition to the first hypothesis. Thus, the objective was to expand the interpretation of the requirement analysis beyond what was included in the actual text and specification to allow for a more implicit analysis that ultimately could enable an interpretation similar to a human's.

To validate this hypothesis, the feasibility of a contextual consideration and interpretation approach was used as a criterion and cross-checked with the evaluation conducted by a human. Through the cross-checks, the accuracy and precision could also be determined.

11.1.4 Hypothesis 4

The complexity of a requirement specification can be quantified based on the defined structure.

The fourth hypothesis is the first that addresses the complexity of a system development process as it targets the quantification of complexity based on the structures that are elicited through the developed approaches (validated in the first three hypotheses). Thus, Hypothesis 4 was validated with a case study that shows the applicability and possibility for quantification. Furthermore, as a criterion for the actual quantification of complexity, mathematical reasoning was used, as well as existing criteria, such as Weyuker's criteria (1988), amongst others.

11.1.5 Hypothesis 5

The complexity of a requirement specification can be quantified in a general way.

Building on the fourth hypothesis, Hypothesis 5 moves beyond the structure and assess the possibility for complexity quantification and its impact in a general way. This feasibility assessment was partially conducted through literature research as well as the case studies in this dissertation.

As a criterion for the fourth hypothesis, the feasibility and general complexity quantification possibilities were used also in combination with the above-mentioned criteria for example. In general, the fourth and fifth hypotheses are tightly connected, and since the latter builds upon the former, a rejection of the fourth would likely also mean a rejection of the fifth. Overall, the fifth hypothesis contributes mainly to future expansion possibilities and thus forms a major part of the last chapter based on the implications and insights gained.

11.1.6 Hypothesis 6

The complexity of requirements and the one of the system can be linked.

The sixth hypothesis attempts to link the results of the fourth and fifth with the actual system and thus targets the discovery of a direct casualty or correlation between the complexity levels of the two sides. This link was planned to be assessed through case studies that test the described connection. To achieve this, the discovery of correlation and or causation was used as a criterion. It would have been deemed acceptable if connections could be defined on a conceptual level through focus groups, if necessary.

Because of the general nature and the wide range of possible factors to consider, the sixth hypothesis was and is the most ambitious of the list. Not only does this hypothesis require a large amount of data to validate, but it also targets the discovery of connections that are affected by a multitude of factors that could be uncontrollable in the worst case. Thus, this hypothesis depends on the availability of the right case studies.

11.1.7 Hypothesis 7

A higher level of requirement complexity increases the potential development effort/costs.

The seventh and last hypothesis addresses the effect of the quantification approaches defined in the previous statements. By targeting the discovery of the nature of the implications, applicability was attempted to be created also in regard to real-world use. To test this hypotheses, case studies assessing the correlation of the metrics were supposed to be used in combination with focus groups to define the effecting possible dynamics to be considered. Hence, the criterion was the existence of positive correlations and their effects on the system development in a quantifiable way. Conceptual results for the effects on the process were deemed acceptable due to the abundance of variables that play a role in a development process.

With all the approaches and criteria defined, the validation was evaluated and documented below.

11.2 Hypotheses Support or Rejection and Limitations

For a summary of the outcome of the validation, Table 11.1 provides the results for all hypotheses. The results were divided into "Supported," if the validation is considered successful; "Partially Supported," if the validation criteria were not able to be completely satisfied but show tendencies that imply the possible satisfaction in the near future; and "Not Supported," if the validation criteria were impossible to be satisfied in general or only at this point in time. Future expansion and possibilities for improvements due to current limitations are outlined in Section 11.3 and further expanded in the last chapter.

Table 11.1 - Research Hypotheses Results

#	Hypothesis	Outcome
1	Requirement text can be categorized and structured based on contextual and explicit content.	Supported
2	Structure and or networks can be derived from categorized requirement texts and content.	Supported
3	A structure and frame for contextual interpretation and reasoning of requirements can be defined.	Supported
4	The complexity of a requirement specification can be quantified based on the defined structure.	Supported
5	The complexity of a requirement specification can be quantified in a general way.	Part. Supported
6	The complexity of requirements and the one of the system can be linked.	Not Supported
7	A higher level of requirement complexity increases the potential development effort/costs.	Supported

11.2.1 Hypothesis 1

As discussed above, the first hypothesis was to be addressed through the feasibility of the categorization of the requirement text through implicit and explicit content. Part of this hypothesis, namely the implicit connections and classification, was shown only on a conceptual level in Chapter 6 due to the reliance on the knowledge base, which was not sufficiently available for any example at the time of this dissertation. Thus, the part of the implicit categorization has to be considered partially supported since a controlled application could not be conducted. The concepts presented, including the created and tested reasoner approach that worked with an ontology as a foundation, point towards the possibility of conducting classifications and implicit connection definitions, as also outlined and demonstrated manually with the DC airplane family example (see Figure 6.12).

For the explicit aspects of the hypothesis, the first case study, using the Skyzer UAV, showed the function and possibility of the classification of the requirement content based on entities. With the presented metrics, the Recall, Precision, and F-Score (for the entity layer in this case), the example showed that the automatic categorization is possible in a valid form, and thus, this hypothesis is supported based on the human checks and presented metrics.

It has to be noted that the limitations described in Chapter 7 also apply to the hypothesis here. Since only one controlled case study could be conducted, the applicability and support of the hypothesis do not mean that it is universally applicable and or transferrable. Since numerous factors affect the applicability, as the second case study has shown, applicability and success depend on the input quality, as well as objectives. Yet, these limitations do not invalidate the support of the hypothesis since even unsuitable datasets and requirement specifications can be improved to fit the current version of the approach. Furthermore, future expandability and improvements are possible and likely, as described in the last chapter.

11.2.2 Hypothesis 2

With the support of the first hypothesis, the validation of the second one was possible by extension. Also conducted and presented in the first case study (and further specified in the second study), the network derivation based on the elicited elements in the requirements is also considered supported. This was addressed by structuring the identified entities in accordance with the rules and patterns to create the connections and sequences (e.g., Figure 6.3). As for the validation, the same approach applies as for the first hypothesis, with the human checks being used to determine the Precision, Recall, and F-Score, this time for the structural network and connection definition. Hence, the demonstrated numbers validate the hypothesis and allow it to be considered supported. It also has to be noted that the achieved accuracy and low false positives far exceed the results achieved by Ferrari & Gnesi (2012), for example, as shown in Chapter 4.

Like the first hypothesis, the second one also inherits the limitations from the case study and is thus not considered universal yet. Due to the high number of possible edge cases when it comes to language and text structure, it cannot be claimed that the defined rules and patterns cover all possible cases and are always valid. Yet, based on the metrics used to assess the approach, we argue that the approach is valid and supports the hypothesis under the given conditions. Furthermore, the absence of ambiguity in text, for example, which is one major obstacle, can always be addressed in preparation for the approach, which would solve a considerable amount of issues and hence further supports the approach. Human-in-the-Loop applications are also a possible solution to consider if further improvement of the algorithm cannot be addressed right away.

11.2.3 Hypothesis 3

The third hypothesis was only possible to be validated on a conceptual level since large enough case studies were not available. Similar problems were also encountered by other research projects (Sinha & de Weck, 2016). In the case of this dissertation, the problem results from the fact that not only would the full-scale validation have required an existing specification and or project but also a sufficiently large knowledge base to use for the interpretation and context definition. As a result, the concept shown in Chapter 6 was validated through reasoning and small scale tests (e.g., reasoners to identify context connections).

We consider Hypothesis 3 supported since the concept turned out to be feasible, and it was also implemented on a small-scale to show and validate the function (see Figure 6.10), for instance. Thus, the conceptual function does suffice as validation for this dissertation, but the limitations in the next paragraph have to be kept in mind.

For one, the limited scale of the shown concepts has to be considered a limitation. Since scalability also affects the knowledge base and context data, it cannot be assumed that while the approach works on a small scale, large-scale applications would behave correctly as well. Since a growing knowledge base means more possible inference points and connections, limits have to be introduced at some point that cannot be tested just yet.

Also, while the function was shown with a created and valid ontology, it cannot be assumed that all ontologies allow for the same type of processing. It is possible, for example, that an ontology is constructed in a way that does not allow for easy or correct inference finding, which is also something that needs to be tested with more samples than what was available. Furthermore, the quality of the input (in this case, the knowledge base) also has an impact on the quality of the results, similar to the NLP algorithm.

All in all, the third hypothesis is considered supported with the described limitations.

11.2.4 Hypothesis 4

The first hypothesis regarding complexity addressed the quantification based on the defined structure. The work for this hypothesis was presented in Chapter 7 and validated according to the aforementioned criteria and approach. With the usable and selected metrics from said chapter, applicability was demonstrated in the first and third case studies. For the application to actual problems and feasibility thereof, the first case study provided the necessary data, and the third case study showed more applicability concepts from an analogy perspective. With the numbers and result quality shown in the first case study and the additional aspects from the third, this hypothesis is considered supported.

Despite the support of the hypothesis, the application of the quantification metrics is not universal just yet, and some limitations exist. For instance, the different metrics, while having been shown to be applicable and useful, do not allow for the clear deduction of effects just yet. This disconnect stems from the analogies made in the third case study and the variety of factors that can affect the networks used as a foundation for the quantification. Thus, while the metrics have been successfully applied and quantified the complexity based on the defined structure, the connections to specific RE tasks, for instance, have to be addressed in the future. Also, comparisons across projects are deemed not possible yet, unless the projects share a high degree of feature/attribute similarity. As shown in the second case study, different projects, while quantifiable individually, do not easily allow for comparison due to each having a different size, for example. Hence, the option to add calibration factors, either mathematically defined or empirically, is most likely necessary when cross-project comparisons are to be attempted.

Yet, despite the limitations, the fourth hypothesis is considered supported since the quantification and use of complexity metrics based on the elicited structure have been successfully demonstrated and applied.

11.2.5 Hypothesis 5

The fifth hypothesis is the first that was not possible to be fully supported. This lack of complete support is a direct consequence of the limitations mentioned for the fourth hypothesis. Since the structure that is elicited from the requirements does also include the structure of said statements in addition to the entity/NLP layer, a complexity assessment for the requirements is possible, and the same support explanation as for the fourth hypothesis applies. Yet, the limitations described above do not allow for validation on a general level. This is due to the fact that while we can now quantify the complexity of the requirement structure, we cannot yet deduce how the requirement specification, in a general way, will behave regarding its complexity.

Since the structure of the requirements is a characteristic of the specification, the quantification methods do apply, but it is possible, for instance, that the overall specification develops different dynamics. Even if such different dynamics were to be observed regarding their implications and effect on the RE process, the structure of the requirements would define the minimum value since the overall complexity would be a sum or culmination of various factors. This is analogous to the problem complexity concept by Salado & Nilchiani (2014), and we argue that this analogy indicates a high potential for success and thus, an overall specification metric should be evaluated further in future research steps. Such future steps could also include possible extension of the system/requirement complexity to system development complexity to account for the additional consideration throughout the process.

Hence, while it is not possible to deduce the effect of topological or structural complexity yet regarding general or reciprocal dynamics of the requirements, the quantification of the structure does allow for insights that add to the general assessment and partial support of the hypothesis is possible. For more details about the steps required to research the missing connections, Chapter 12 discusses the necessary objectives.

11.2.6 Hypothesis 6

The sixth hypothesis, being the most ambitious, ended up not being possible to validate/support at this point. The failure to support the hypothesis and consequently having to reject it was in part also influenced by the partial support for the fifth hypothesis. As already described by the limitations of the previous hypothesis, since the structural complexity is only one part of the overall complexity, connecting it to the system in general was not possible. This impossibility is also partially due to the fact that support and validation of the sixth hypothesis would have required a significant amount of data to prove a correlation that is not limited to a specific case. Since this amount of data was not available, which also poses a limitation for Hypothesis 3, no insights and analyses could be conducted to provide any foundation for the sixth hypothesis.

Furthermore, through the course of the research in this dissertation, various aspects emerged that make a direct connection between requirement and system complexity unlikely. These aspects shall be described in the following paragraphs.

First, the system development process goes through various stages at which decisions are made that have to be considered coincidental due to insufficient information. For instance, when design decisions are made throughout the process, not only one solution can satisfy a set of requirements. It has to be considered that there is a multitude of possible solutions that can satisfy a given set of requirements. Yet, not every solution that satisfies a set of requirements will have the same level of complexity; some will be better, some worse. Thus, a direct correlation between requirement complexity and system complexity is unlikely, if not impossible, due to the multitude of factors playing a role.

Second, the decisions made during the development process all influence each other. For example, the choice to go with a specific technology will affect other parts and components of the system, which will, in turn, yield a different solution that potentially satisfies the requirements. Hence, the decision process has to be considered as a more or less coincidental and arbitrary tree with an unpredictable number of branches. Figure 11.1 illustrates such a tree conceptually. Given that the requirement set is separate from said tree, a direct connection to the end solution cannot be made.

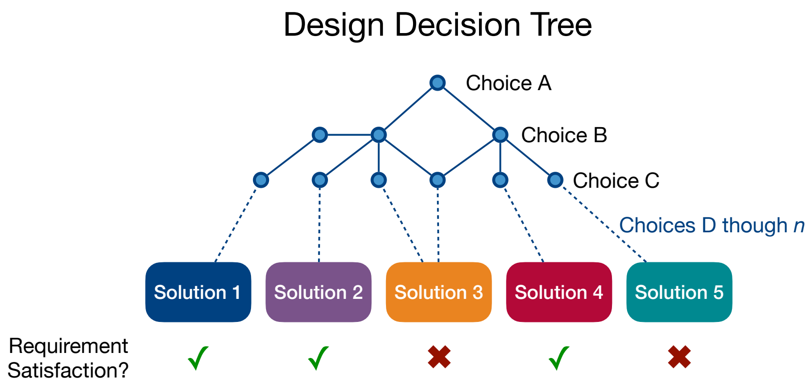

FIG. 11.1 - HYPOTHETICAL DESIGN DECISION TREE SHOWING POSSIBLE OUTCOMES

The described missing connection also has been conceptually tested by combining requirement complexity quantification, as shown in Case Study 1, with trade space analyses. Expanding the trade space by one axis and having said axis represent the requirement complexity metric showed that it would not add information that helps find better solutions as the values for multiple configurations shared the same requirement complexity.

Lastly, the complexity introduced throughout the process has to be considered compounding and possibly even multiplies in some cases. This is a direct conclusion of the circumstances described above, and since multiple factors contribute to the complexity of the system, reciprocities and reinforcing relationships are possible. As a result, a direct connection between the requirements and the system becomes even more unlikely since these compounding and potentially dynamic connections would make the discovery of correlations difficult and hard to verify, which affects causality relationships even more.

All in all, it has been concluded that at this point in time, with the given approach and framework generated, no direct connection between the complexity of a requirement set and a resulting system can be specified. It is possible that such a connection, as it stems from the inherent definition of complexity, might not be possible to define at all without removing complexity, at which point the hypothesis becomes obsolete. Yet, approaches such as Artificial Intelligence or Machine Learning could be contenders to address this problem as well.

11.2.7 Hypothesis 7

The last hypothesis, addressing the effect of requirement complexity on the potential system development effort and or cost, is considered supported based on the results of the experiment used in the third case study. As shown in Chapter 10, a positive correlation exists between the majority of the chosen metrics and a higher time effort required from the subjects. As per the connections of this effort/time to cost described in the last section of Chapter 7, we argue that the correlation between those metrics will affect the system development as per the correlations and as shown by (Valerdi, 2008). Thus, the hypothesis is supported.

The correlations found in the third case study show that a higher level of complexity, as per the used metrics, will require higher effort and more time from the humans working with a specific constellation. Hence, a set of requirements with higher complexity will potentially have a longer development cycle, more errors, and or produce lower quality. Since these factors are directly related to cost, monetary implications are a direct result.

Given that the results of the case study are related to the requirement complexity via the analogous connection described, no direct implications can be defined as to what task would be affected and how. Yet, since the hypothesis only states a general correlation, it can be supported based on the results. Also, since the implications of requirements in the systems development process are manifold, the exact effect manifestation is to be defined in future research.

Now, with the explanations above, which state that the requirement complexity is one part contributing to the potential system complexity, another to consider is that, while the effect has been shown in the case study to exist, the magnitude of the effect on the development process cannot be gauged yet. With more than one factory introducing complexity, the share of each can vary from project to project or system to system. Thus, while the seventh hypothesis is supported, its implications for real-life cases and applications are subject to further research.

It also has to be noted that the conflicting results discovered by other researchers, for example, Sinha & de Weck (2016), further underline the variety and likely diverse effects that the discovered connections have. One manifestation of such variety is the subjectivity found in the third case study, which merits further investigation of calibration factors, potentially even for established metrics, such as the one by Sinha & de Weck (2016).

With all the hypotheses addressed, the last part left in this chapter is a discussion and also a general description of the limitations of the presented results, which is addressed in the last section below.

11.3 Discussion and Limitations

With all the hypotheses addressed and concluded in the previous sections, the last aspect to address is to discuss the outcome and also mention some general limitations of the presented work, which are both addressed in the following subsections individually.

11.3.1 Discussion of the Results

When we look at the results as a whole, as shown in Table 11.1, we see that overall, the hypotheses could be supported in most cases. The concept and approach generated showcased the feasibility and applicability of most of the goals that were set with and through the hypotheses. In addition, even the hypotheses that could not be fully supported show promising opportunities for future research. Where the limitations in the next subchapter were met, new and previously undiscovered possible solutions, such as the application of Machine Learning, can be seen and merit further investigation.

In addition, what has been shown throughout the validation chapter, is a high degree of dependency that also becomes clear, looking back at Figure 5.2. Due to the reliance of some hypotheses on the results of others, their validation was impacted by any restrictions that are inherited through dependencies. Yet, this was considered in the approach development and thus mitigated within the limitations described below (e.g., through the modularized approach).

For the overall results, what has to be included in this discussion is the fact that while the results achieved are valid and allow for the support of the mentioned hypotheses, we cannot claim overall validity and universal applicability yet. While this was never the goal of the research presented in this dissertation, it is important to note. Due to the multitude of variables that not only affect the system development process as well as the requirements and their management/engineering discipline, many more rare cases can be imagined where the developed approach and concepts would require adaptation or changes. Thus, the modularity conceptualized and shown in Figure 5.5 also retains the flexibility to modify the generated approaches where needed in the future.

Overall, the hypotheses validated and concluded in this chapter can be considered successful, given that all but one could be supported, especially since the one exception was the most ambitious hypothesis to begin with. Yet, the limitations that were and had to be introduced throughout the research have to be kept in mind and thus are listed and explained in the last subsection.

11.3.2 Limitations of the Hypotheses and Research

In total, four limitations currently exist for the developed approaches and presented research results: 1) limited universal validity; 2) restrictions to the transferability of the approaches; 3) limits of some approach capabilities; 4) availability limitations regarding input knowledge and data (specifically the knowledge base). These four aspects are explained below.

The first limitation was already touched upon above, namely the lack of universal applicability of the approaches when it comes to all possible real-world applications. Since the topic and the developed approaches deal with systems development projects that are not exclusive to one type of product, company, organization, or even industry, covering all possible eventualities is difficult and, in some cases, impossible. Thus, assumptions were made throughout the development of the approaches, such as the more or less strict adherence to requirement standards for the NLP input, in order to achieve controllable case studies and datasets that allowed for validity checks and verification. The one case where the limitation of the presented research becomes clear is the second case study. Because of the high degree of variability in the dataset of Case Study 2, the application of the approach, while possible, would not have yielded any conclusive results that are worth interpreting due to their lack of validity. These limitations also apply to general projects and other system development processes that are possible application contenders. While the approach might be usable in many cases, the actual validity and, more importantly, comparability of results has to be ensured before attempting and using the presented research and framework.

The second limitation, the lack of transferability, overlaps in part with the first one. Yet, when we look at the transferability of the presented research, we see that some choices were made throughout the process that limit the transferability possibilities. For example, the choice of spaCy limits the NLP application to the languages that the core libraries are trained on. Thus, while a transfer of the approach to different environments and circumstances is possible and likely to function correctly, adjustments might be necessary and should be carefully examined upon consideration of the assumptions described in the respective chapters. This limitation also applies to the complexity, for example, as the choices and assumptions in Chapter 7 outline.

The third limitation, which describes the limits of some of the approach capabilities, goes hand in hand with the future expansion possibilities in the next chapter. Because the Ph.D. program that this dissertation is a result of was limited in time and scope, certain decisions had to be made that ultimately limited the capabilities of the developed approach. For instance, while NLP can address textual requirements based on their sentence and content structure, other artifacts, such as tables and figures, might require an extension of the approach since these artifacts do not fit the patterns and rules defined so far. The same aspects apply to non-standard requirements and requirements that are not mature enough before procurement begins (Hooks, 2001). Thus, while the framework and approach that were created are fully functional and valid, they still have great potential for expansion and continued research. For a comprehensive list of future research possibilities (see Section 12.2).

The last limitation concerns resource problems regarding the knowledge base implementation (see Section 6.5 ff.). While the concept could be demonstrated, the actual application could not be tested due to no sufficiently large datasets being available. There are companies that offer ontology creation for a fee, but those offerings were outside the budget of the research project, and they would have required precise tailoring to a case study to enable applicability. Such a setup was not possible to be arranged, which is why the possibility of acquisition was not chosen, and the concept demonstration was provided instead. Yet, this choice means that the actual application is subject to further refinements and improvements to enable the conceptually valid function on a larger scale and in different environments.

The limitations above conclude the validation chapter and also the core work/contribution of this dissertation. The last chapter provides a summary that lists the specific results, as well as future work opportunities, before concluding the dissertation.

Bibliography of Chapter 11

Ferrari, A., & Gnesi, S. (2012, 24-28 Sept. 2012). *Using collective intelligence to detect pragmatic ambiguities.* Paper presented at the 2012 20th IEEE International Requirements Engineering Conference (RE).

Hooks, I. F. (2001). Managing requirements. *NJIT Requirements Engineering Handout*, 1-8.

Salado, A., & Nilchiani, R. (2014). The Concept of Problem Complexity. *Procedia Computer Science, 28*, 539-546. doi:https://doi.org/10.1016/j.procs.2014.03.066

Sinha, K., & de Weck, O. L. (2016). Empirical Validation of Structural Complexity Metric and Complexity Management for Engineering Systems. *Systems Engineering, 19*(3), 193-206. doi:10.1002/sys.21356

Valerdi, R. (2008). *The Constructive Systems Engineering Cost Model (COSYSMO): Quantifying the costs of systems engineering effort in complex systems*. Saarbrücken, Germany: VDM Verlag.

Weyuker, E. J. (1988). Evaluating software complexity measures. *IEEE Transactions on Software Engineering, 14*(9), 1357-1365. doi:10.1109/32.6178

CHAPTER 12: CONCLUSION AND OUTLOOK

"It is difficult to predict, especially the future."

Niels Bohr

We have seen in the previous chapters that this dissertation has addressed a variety of fields, ideas, and topics. To provide a final overview for the content, this last chapter will provide a summary of the results and specific contributions in Section 12.1. Building on the contributions, Section 12.2 then extends the presented research to assess possible future opportunities, including their values. Lastly, the final section (12.3) provides a conclusion for this dissertation.

12.1 Summary, Results, and Contributions

The presented dissertation addressed in its core the quantification of structural complexity factors regarding requirements and RE of the system development process. In order to measure and gauge said complexity, the research attempted and achieved the elicitation of structure from textural requirements that were used as a foundation. Building on this foundation, the complexity inherent to the requirement structure, hierarchy, and architecture could be calculated. In addition, a concept for the inclusion and consideration of context and implicit connections that are not explicitly stated was created.

With the structure of and within the requirements as well as its complexity, the effects of different metrics were related to the effort of humans regarding integration tasks and general operations. The conducted research showed a strong correlation between four of the chosen complexity metrics and the time that humans needed to complete integration tasks. This correlation is indicative of the effects that can be expected in the system development & design process regarding the work with requirements therein.

The main result of the dissertation is the framework developed in Chapter 5 and shown again below without annotations:

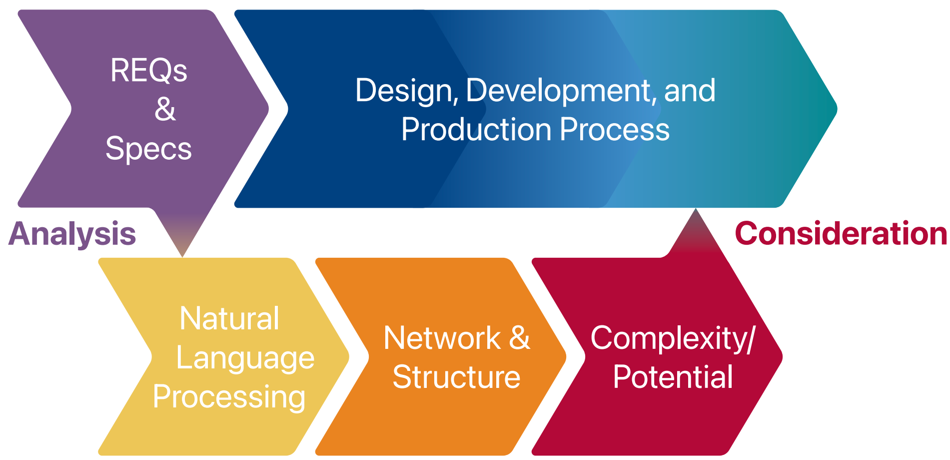

FIG. 12.1 - STRUCTURAL COMPLEXITY OF SYSTEM REQUIREMENTS FRAMEWORK

The depicted framework allows for the analysis and consideration of system requirements regarding their complexity starting at the beginning of the process without relying on elements of the system or its design. Thus, this framework, with its included components, such as the NLP algorithm, the context consideration approach, and the complexity metrics, poses the main contributions of this dissertation. Put succinctly, the dissertation produced the following six results:

1. Generation of a functional and accurate NLP algorithm that elicits three structural structure levels from textual requirements: the Hierarchy Structure, the Requirement Structure, and the Entity/NLP Structure
2. Development of an implementation framework for the consideration of contextual and implicit connections by using a knowledge base in the form of an ontology

3. An interactive representation of the network structures underlying the requirements to allow for graphical analysis and understanding
4. Quantification of the complexity within the requirements as well as the entities through the application of spectral theory metrics, cyclomatic approaches, and general attribute-based measures (see density and Load *L*)
5. Discovery and analysis of strong positive correlations between the complexity quantification metrics and the effects thereof on human tasks as well as effort that can be related to development process aspects, such as cost and time
6. Integration of the points above (with the exception of 2.) into a holistic and automatic approach that uses a requirement specification as input

In one sentence, the presented research has shown the possibility of eliciting structure from textural requirements, then using said structure to quantify the complexity of the requirements, and finally correlating this complexity positively to human effort and time regarding the handling and work with the requirements. As such, this dissertation contributes to all three fields (complexity, RE, NLP) initially assessed in the literature reviews as well as the introduction.

To the field of complexity science, the presented research contributes an application of existing methods and ideas to a previously not considered field: requirements engineering. Thus, the contributions of this dissertation open the door for complexity science to include and further expand in the direction of requirements and in return, benefit from the insights generated by such an expansion.

To the field of RE, the dissertation at hand contributes an approach that can analyze the complexity, structure, as well as architecture of requirements from the beginning of the process onward. This approach can also be used iteratively and recursively and is not limited to a single application but may even benefit from repeated assessments. This possibility also makes the approach and framework suitable for Agile, as repeated application is a key focus in these types of environments and methods (see Section 3.2).

To the field of NLP and NLP4RE, the presented work contributes an automatic approach that requires little to no human interaction and can consider context as well as implicit links. Thus, by enabling all of the shown analyses in conjunction with the two previously mentioned fields, the NLP and NLP4RE space gained valuable interfaces to further expand as well. Also, the created approach to elicit structure from requirements is unique based on the presented literature research and thus, poses an individual and separate contribution.

Lastly, the research efforts have provided additional, unplanned insights. For one, the subjectivity aspects mentioned and seen in the last case study are potentially of interest due to their recurrence and consistency. Also, the applicability and usability of ontologies as context knowledge bases, while not planned in the beginning, showed valid function and thus could be considered for other approaches and fields as well. Lastly, the graphical representations, while not initially planned, turned out to be versatile and extremely useful for demonstration purposes, which is why further developments in this direction shows merit as well.

12.2 Future Work Opportunities and Possibilities

As is so often, the contributions produced in this dissertation, while solving problems, have generated new ones as well as new ideas. Some of these ideas are direct results of the limitations discussed in the previous chapter, but not all of them.

First, the created NLP approach, while valid and verified within the chosen frame, has to be evaluated on a larger scale. Since the unpredictability of language can manifest itself in various forms, it is not possible to claim consideration of all eventualities on the scale presented.

Therefore, expanding and improving the NLP algorithm is one valuable point to pursue in the future. Such improvements and expansion can also include other approaches, such as Machine Learning and or Artificial Intelligence (including other tools as mentioned in Chapter 6), since the current setup of the algorithm relies on patterns and rules which could easily be expanded and or amended by machine-identified additions. Similar opportunities exist for the knowledge base and ontology aspects since they are subject to similar scalability challenges.

Second, the applied complexity metrics, while valid and reasonable, can be further extended to include tests in other directions. As shown by the abundance of topics and directions in Chapter 2, there are various contenders that could prove useful and provide novel insights based on the foundation of requirements. These extension possibilities also include the consideration of combination approaches, as also mentioned in Chapter 11. By integrating the requirement complexity and or combining it with other approaches, potentially even causality assessments, the effect and understanding of the dynamics can be significantly improved.

Third, the biggest limitation of the presented research, which also led to the rejection of the sixth hypothesis, is the biggest opportunity for future research. While a link and causal connection from requirements to actual system complexity is difficult to define, due to the factors discussed in Chapter 11, causalities of some sort are not impossible per se. Thus, revisiting the foundations of those connections and potentially assessing where impacts and effects exist is the most promising aspect of potential future directions. If a causal relationship between the complexity of requirements and the one of a system could be defined, not only would the understanding of the dynamics greatly be improved, but also optimization and improvement possibilities would be enabled. Such possibilities could make the consideration shown in Figure 12.1 a major driver of the system development process, and complexity therein could be assessed, managed, and hedged much more effectively from the beginning.

Lastly, the effects of what could be subjectivity plus the failure of the second case study point to another problem that is worth exploring: lateral or cross-project/cross-system comparability. If subjective as well as system/project-specific aspects play a role when it comes to complexity and its implications, comparisons and transfer of concepts become challenging. In such cases, calibration factors can be used, but their definition can be problematic or rely on large datasets for empiric definition. Thus, evaluating the difference drivers and what attributes make transfer and cross-connections difficult is another promising extension of the research efforts. It is possible that through the exploration and discovery of difference factors, even other metrics and approaches could be enhanced and amended.

12.3 Conclusion

The work and research presented in this dissertation set out to address a problem that brought together multiple research fields and topics. By approaching the problem systematically and methodologically, we argue that good results were produced, and based on the support of all but one hypothesis, an overall successful research project comes to a close. Besides enabling and showing multiple different new approaches and concepts, as well as novel approaches and a framework, the presented research also contributed valuable insights regarding current limitations and boundaries.

The shown future possibilities and opportunities are as multifaceted as the problems discussed in this dissertation, maybe even more so. Most likely, each of the outlined possible path ideas can fill a whole dissertation by itself, and other researchers and colleagues might see many more possible trajectories in addition to the ones outlined above. We encourage everyone reading this dissertation to consider topics, ideas, and thoughts for further research so that the work and efforts put into this research can support progress beyond the results and insights produced so far.

APPENDIX A - LIST OF PUBLICATIONS DERIVED FROM THIS DISSERTATION

Peer-Reviewed Journal Publications

Vierlboeck, M., Nilchiani, R. R., & Edwards, C. M. (2021). The Pandemic Holiday Blip in New York City. *IEEE Transactions on Computational Social Systems, 8*(3), 568-577. doi:10.1109/TCSS.2021.3058633

Submitted Journal Publications

Vierlboeck, M., Nilchiani, R. R., Ganguly, A., & Edwards, C. M. Evaluating the Tipping Point of a Complex System: The Case of Disruptive Technology. *Wiley Systems Engineering.*

Vierlboeck, M., Nilchiani, R. R., & Blackburn, M. Natural Language Processing to Assess Structure and Complexity of System Requirements. *IEEE OJSE.*

Conference Publications and Presentations

Dunbar, D., Vierlboeck, M., & Blackburn, M. (2023). *Use of Natural Language Processing in Digital Engineering Context to Aid Tagging of Model*. Paper presented at the 2023 IEEE International Systems Conference (SysCon).

Xiao, L., Zhao, G., Vierlboeck, M., & Nilchiani, R. R (2023). *A Framework on Early Decoupling Level Metric Assessment based on NLP4RE*. Paper presented at the 2023 International Conference on Systems Engineering Research (CSER).

Vierlboeck, M., Nilchiani, R. R., Ganguly, A., & Edwards, C. M. (2022, 24-26 Oct. 2022). *Requirements Engineering for the Development of Disruptive Systems Engineering Innovations.* Paper presented at the 2022 IEEE International Symposium on Systems Engineering (ISSE).

Vierlboeck, M., Dunbar, D., & Nilchiani, R. R. (2022, 25-28 April 2022). *Natural Language Processing to Extract Contextual Structure from Requirements.* Paper presented at the 2022 IEEE International Systems Conference (SysCon).

Vierlboeck, M., & Nilchiani, R. R (2021). *Complexity at the Requirement Stage – A Novel Paradigm for System Development.* Paper presented at the 2021 SERC Doctoral Student Forum.

Vierlboeck, M., & Nilchiani, R. R. (2021, 13 Sept.-13 Oct. 2021). *Requirement Engineering in the Age of System and Product Complexity – A Literature Review.* Paper presented at the 2021 IEEE International Symposium on Systems Engineering (ISSE).

Pugliese, A., Nilchiani, R. R., & Vierlboeck, M. (2021). *Structural Complexity Analysis to Evaluate Technical Risk in Defense Acquisition*. Paper presented at the 2021 Acquisition Research Program.

Vierlboeck, M., Nilchiani, R. R., & Edwards, C. M. (2020, 12 Oct.-12 Nov. 2020). *Systems Approach to Localize Tipping Points for the Emergency Services in Face of the COVID-19 Pandemic.* Paper presented at the 2020 IEEE International Symposium on Systems Engineering (ISSE).

Arxiv and Pre-Print Publications

Edwards, C. M., Nilchiani, R. R., Ganguly, A., & Vierlboeck, M. (2022). Evaluating the Tipping Point of a Complex System: The Case of Disruptive Technology. doi:10.36227/techrxiv.21126292.v1

Vierlboeck, M., Nilchiani, R. R., & Blackburn, M. (2022). Natural Language Processing to Assess Structure and Complexity of System Requirements. doi:10.36227/techrxiv.20818984.v1

Vierlboeck, M., Lipizzi, C., & Nilchiani, R. R. (2022). Natural Language in Requirements Engineering for Structure Inference – An Integrative Review. doi:10.48550/arXiv.2202.05065

Vierlboeck, M., & Nilchiani, R. R. (2021). Effects of COVID-19 Vaccine Developments and Rollout on the Capital Market – A Case Study. doi:10.48550/arXiv.2105.12267

Vierlboeck, M., Nilchiani, R. R., & Edwards, C. M. (2020). The Easter and Passover Blip in New York City: How Exceptions Can Cause Detrimental Effects in Pandemic Times. doi:10.1101/2020.04.14.20065300

APPENDIX B - NLP DATASET AND SOURCE LIST

ADDITIONAL REFERENCES FOR CHAPTER 4

Abad, Z. S. H., Gervasi, V., Zowghi, D., & Barker, K. (2018, 21-21 Aug. 2018). *ELICA: An Automated Tool for Dynamic Extraction of Requirements Relevant Information.* Paper presented at the 2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE).

Aguilera, C., & Berry, D. M. (1990). The use of a repeated phrase finder in requirements extraction. *Journal of Systems and Software, 13*(3), 209-230. doi:10.1016/0164-1212(90)90097-6

Al-Safadi, L. A. (2009). Natural language processing for conceptual modeling. *International Journal of Digital Content Technology and its Applications, 3*(3), 47-59.

Alfonso Hoyos, J. P., & Restrepo-Calle, F. (2018). Fast Prototyping of Web-Based Information Systems Using a Restricted Natural Language Specification, Cham.

Ali, N., Gueheneuc, Y., & Antoniol, G. (2011, 17-20 Oct. 2011). Requirements Traceability for Object Oriented Systems by Partitioning Source Code. Paper presented at the 2011 18th Working Conference on Reverse Engineering.

Ali, N., Gueheneuc, Y., & Antonioldupl, G. (2011, 22-24 June 2011). *Trust-Based Requirements Traceability.* Paper presented at the 2011 IEEE 19th International Conference on Program Comprehension.

Ali, N., Jaafar, F., & Hassan, A. E. (2013, 14-17 Oct. 2013). *Leveraging historical co-change information for requirements traceability.* Paper presented at the 2013 20th Working Conference on Reverse Engineering (WCRE).

Ambriola, V., & Gervasi, V. (1997, 1-5 Nov. 1997). *Processing natural language requirements.* Paper presented at the Proceedings 12th IEEE International Conference Automated Software Engineering.

Ambrosio, A. P., Métais, E., & Meunier, J.-N. (1997). The linguistic level: Contribution for conceptual design, view integration, reuse and documentation. *Data & Knowledge Engineering, 21*(2), 111-129. doi:10.1016/S0169-023X(96)00028-6

Andersen, A., & Munch, K. H. (1991). Automatic generation of technical documentation. *Expert Systems with Applications, 3*(2), 219-227. doi:10.1016/0957-4174(91)90150-D

Angelov, K., Camilleri, J. J., & Schneider, G. (2013). A framework for conflict analysis of normative texts written in controlled natural language. *The Journal of Logic and Algebraic Programming, 82*(5), 216-240. doi:10.1016/j.jlap.2013.03.002

Arora, C., Sabetzadeh, M., Briand, L., & Zimmer, F. (2015). Automated Checking of Conformance to Requirements Templates Using Natural Language Processing. *IEEE Transactions on Software Engineering, 41*(10), 944-968. doi:10.1109/TSE.2015.2428709

Arora, C., Sabetzadeh, M., Briand, L., & Zimmer, F. (2017). Automated Extraction and Clustering of Requirements Glossary Terms. *IEEE Transactions on Software Engineering, 43*(10), 918-945. doi:10.1109/TSE.2016.2635134

Arora, C., Sabetzadeh, M., Briand, L., Zimmer, F., & Gnaga, R. (2013, 10-11 Oct. 2013). *Automatic Checking of Conformance to Requirement Boilerplates via Text Chunking: An Industrial Case Study.* Paper presented at the 2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement.

Arora, C., Sabetzadeh, M., Goknil, A., Briand, L. C., & Zimmer, F. (2015, 24-28 Aug. 2015). *Change impact analysis for Natural Language requirements: An NLP approach.* Paper presented at the 2015 IEEE 23rd International Requirements Engineering Conference (RE).

Bajwa, I. S., & Choudhary, M. A. (2012). From Natural Language Software Specifications to UML Class Models, Berlin, Heidelberg.

Bakar, N. H., Kasirun, Z. M., Salleh, N., & Jalab, H. A. (2016). Extracting features from online software reviews to aid requirements reuse. *Applied Soft Computing, 49*, 1297-1315. doi:10.1016/j.asoc.2016.07.048

Barcellini, F., Albert, C., Grosse, C., & Saint-Dizier, P. (2012, may). Risk Analysis and Prevention: LELIE, a Tool dedicated to Procedure and Requirement Authoring, Istanbul, Turkey.

Boddu, R., Guo, L., Mukhopadhyay, S., & Cukic, B. (2004). RETNA: From Requirements to Testing in a Natural Way. *Proceedings. IEEE International Requirements Engineering Conference, 2004*, 262-271. doi:10.1109/ICRE.2004.1335683

Boutkova, E., & Houdek, F. (2011, 29 Aug.-2 Sept. 2011). *Semi-automatic identification of features in requirement specifications.* Paper presented at the 2011 IEEE 19th International Requirements Engineering Conference.

Buchholz, E., Düsterhöft, A., & Thalheim, B. (1997). Capturing information on behaviour with the RADD-NLI: A linguistic and knowledge-based approach. *Data & Knowledge Engineering, 23*(1), 33-46. doi:10.1016/S0169-023X(97)00009-8

Carreño, L. V. G., & Winbladh, K. (2013, 18-26 May 2013). *Analysis of user comments: An approach for software requirements evolution.* Paper presented at the 2013 35th International Conference on Software Engineering (ICSE).

Carvalho, G., Falcão, D., Barros, F., Sampaio, A., Mota, A., Motta, L., & Blackburn, M. (2014). NAT2TESTSCR: Test case generation from natural language requirements based on SCR specifications. *Science of Computer Programming, 95*, 275-297. doi:10.1016/j.scico.2014.06.007

Casagrande, E., Woldeamlak, S., Woon, W. L., Zeineldin, H. H., & Svetinovic, D. (2014). NLP-KAOS for Systems Goal Elicitation: Smart Metering System Case Study. *IEEE Transactions on Software Engineering, 40*(10), 941-956. doi:10.1109/TSE.2014.2339811

da Silva, A. R. (2015). SpecQua: Towards a Framework for Requirements Specifications with Increased Quality, Cham.

Dag, J. N. o., Gervasi, V., Brinkkemper, S., & Regnell, B. (2004, 10-10 Sept. 2004). Speeding up requirements management in a product software company: linking customer wishes to product requirements through linguistic engineering. Paper presented at the Proceedings. 12th IEEE International Requirements Engineering Conference, 2004.

Dalpiaz, F., & Parente, M. (2019). RE-SWOT: From User Feedback to Requirements via Competitor Analysis, Cham.

Daramola, O., Stålhane, T., Sindre, G., & Omoronyia, I. (2011, 30-30 Aug. 2011). *Enabling hazard identification from requirements and reuse-oriented HAZOP analysis.* Paper presented at the 2011 4th International Workshop on Managing Requirements Knowledge.

De Tommasi, M., & Corallo, A. (2006). SBEAVER: A Tool for Modeling Business Vocabularies and Business Rules, Berlin, Heidelberg.

Deeptimahanti, D. K., & Sanyal, R. (2009). An Innovative Approach for Generating Static UML Models from Natural Language Requirements, Berlin, Heidelberg.

Di, F., & Zhang, M. (2009, 11-13 Dec. 2009). *An Improving Approach for Recovering Requirements-to-Design Traceability Links.* Paper presented at the 2009 International Conference on Computational Intelligence and Software Engineering.

Dori, D., Korda, N., Soffer, A., & Cohen, S. (2004). SMART: System Model Acquisition from Requirements Text, Berlin, Heidelberg.

El-Attar, M. (2012). Towards developing consistent misuse case models. *Journal of Systems and Software, 85*(2), 323-339. doi:10.1016/j.jss.2011.08.023

Fabbrini, F., Fusani, M., Gnesi, S., & Lami, G. (2001, 27-29 Nov. 2001). The linguistic approach to the natural language requirements quality: benefit of the use of an automatic tool. Paper presented at the Proceedings 26th Annual NASA Goddard Software Engineering Workshop.

Ferrari, A., dell'Orletta, F., Spagnolo, G. O., & Gnesi, S. (2014). Measuring and Improving the Completeness of Natural Language Requirements, Cham.

Ferrari, A., Spagnolo, G. O., Gnesi, S., & Dell'Orletta, F. (2015). *CMT and FDE: tools to bridge the gap between natural language documents and feature diagrams*. Paper presented at the Proceedings of the 19th International Conference on Software Product Line, Nashville, Tennessee. https://doi.org/10.1145/2791060.2791117

Ferreira, D. d. A., & Silva, A. R. d. (2012, 24-24 Sept. 2012). *RSLingo: An information extraction approach toward formal requirements specifications.* Paper presented at the 2012 Second IEEE International Workshop on Model-Driven Requirements Engineering (MoDRE).

Fliedl, G., Kop, C., Mayr, H. C., Mayerthaler, W., & Winkler, C. (2000). Linguistically based requirements engineering – The NIBA-project. *Data & Knowledge Engineering, 35*(2), 111-120. doi:10.1016/S0169-023X(00)00029-X

Gacitua, R., Sawyer, P., & Gervasi, V. (2010, 27 Sept.-1 Oct. 2010). *On the Effectiveness of Abstraction Identification in Requirements Engineering.* Paper presented at the 2010 18th IEEE International Requirements Engineering Conference.

Génova, G., Fuentes, J. M., Llorens, J., Hurtado, O., & Moreno, V. (2013). A framework to measure and improve the quality of textual requirements. *Requirements Engineering, 18*(1), 25-41. doi:10.1007/s00766-011-0134-z

Gervasi, V., & Zowghi, D. (2005). Reasoning about inconsistencies in natural language requirements. *ACM Trans. Softw. Eng. Methodol., 14*(3), 277–330. doi:10.1145/1072997.1072999

Ghosh, S., Elenius, D., Li, W., Lincoln, P., Shankar, N., & Steiner, W. (2016). *ARSENAL: Automatic Requirements Specification Extraction from Natural Language*, Cham.

Gibiec, M., Czauderna, A., & Cleland-Huang, J. (2010). *Towards mining replacement queries for hard-to-retrieve traces*. Paper presented at the Proceedings of the IEEE/ACM international conference on Automated software engineering, Antwerp, Belgium. https://doi.org/10.1145/1858996.1859046

Giganto, R., & Smith, T. (2008, 13-18 April 2008). Derivation of Classes from Use Cases Automatically Generated by a Three-Level Sentence Processing Algorithm. Paper presented at the Third International Conference on Systems (icons 2008).

Gokyer, G., Cetin, S., Sener, C., & Yondem, M. T. (2008, 26-31 Oct. 2008). *Non-functional Requirements to Architectural Concerns: ML and NLP at Crossroads.* Paper presented at the 2008 The Third International Conference on Software Engineering Advances.

Goldin, L., & Berry, D. M. (1994, 18-22 April 1994). AbstFinder, a prototype abstraction finder for natural language text for use in requirements elicitation: design, methodology, and evaluation. Paper presented at the Proceedings of IEEE International Conference on Requirements Engineering.

Guzman, E., Ibrahim, M., & Glinz, M. (2017, 4-8 Sept. 2017). *A Little Bird Told Me: Mining Tweets for Requirements and Software Evolution.* Paper presented at the 2017 IEEE 25th International Requirements Engineering Conference (RE).

Hajri, I., Goknil, A., Briand, L. C., & Stephany, T. (2018). Configuring use case models in product families. *Software & Systems Modeling, 17*(3), 939-971. doi:10.1007/s10270-016-0539-8

Harmain, H. M., & Gaizauskas, R. (2000, 11-15 Sept. 2000). *CM-Builder: an automated NL-based CASE tool.* Paper presented at the Proceedings ASE 2000. Fifteenth IEEE International Conference on Automated Software Engineering.

Hayes, J. H., Dekhtyar, A., & Sundaram, S. K. (2006). Advancing candidate link generation for requirements tracing: the study of methods. *IEEE Transactions on Software Engineering, 32*(1), 4-19. doi:10.1109/TSE.2006.3

Hotomski, S., & Glinz, M. (2019). GuideGen: An approach for keeping requirements and acceptance tests aligned via automatically generated guidance. *Information and Software Technology, 110*, 17-38. doi:10.1016/j.infsof.2019.01.011

Hussain, I., Ormandjieva, O., & Kosseim, L. (2007, 11-12 Oct. 2007). Automatic Quality Assessment of SRS Text by Means of a Decision-Tree-Based Text Classifier. Paper presented at the Seventh International Conference on Quality Software (QSIC 2007).

Jain, P., Verma, K., Kass, A., & Vasquez, R. G. (2009). Automated review of natural language requirements documents: generating useful warnings with user-extensible glossaries driving a simple state machine. Paper presented at the Proceedings of the 2nd India software engineering conference, Pune, India. https://doi.org/10.1145/1506216.1506224

Johann, T., Stanik, C., B, A. M. A., & Maalej, W. (2017, 4-8 Sept. 2017). *SAFE: A Simple Approach for Feature Extraction from App Descriptions and App Reviews.* Paper presented at the 2017 IEEE 25th International Requirements Engineering Conference (RE).

Jurkiewicz, J., & Nawrocki, J. (2015). Automated events identification in use cases. *Information and Software Technology, 58*, 110-122. doi:10.1016/j.infsof.2014.09.011

Kamalrudin, M., Hosking, J., & Grundy, J. (2017). MaramaAIC: tool support for consistency management and validation of requirements. *Automated Software Engineering, 24*(1), 1-45. doi:10.1007/s10515-016-0192-z

Kchaou, D., Bouassida, N., Mefteh, M., & Ben-Abdallah, H. (2019). Recovering semantic traceability between requirements and design for change impact analysis. *Innovations in Systems and Software Engineering, 15*(2), 101-115. doi:10.1007/s11334-019-00330-w

Korner, S. J., & Brumm, T. (2009, 14-16 Sept. 2009). *RESI - A Natural Language Specification Improver.* Paper presented at the 2009 IEEE International Conference on Semantic Computing.

Landhäußer, M., Körner, S. J., & Tichy, W. F. (2014). From requirements to UML models and back: how automatic processing of text can support requirements engineering. *Software Quality Journal, 22*(1), 121-149. doi:10.1007/s11219-013-9210-6

Landhäußer, M., Korner, S. J., Tichy, W. F., Keim, J., & Krisch, J. (2015, 24-24 Aug. 2015). *DeNom: a tool to find problematic nominalizations using NLP.* Paper presented at the 2015 IEEE Second International Workshop on Artificial Intelligence for Requirements Engineering (AIRE).

Lee, B.-S., & Bryant, B. R. (2002). *Contextual natural language processing and DAML for understanding software requirements specifications*. Paper presented at the Proceedings of the 19th international conference on Computational linguistics - Volume 1, Taipei, Taiwan. https://doi.org/10.3115/1072228.1072352

López, C., Codocedo, V., Astudillo, H., & Cysneiros, L. M. (2012). Bridging the gap between software architecture rationale formalisms and actual architecture documents:

An ontology-driven approach. *Science of Computer Programming, 77*(1), 66-80. doi:10.1016/j.scico.2010.06.009

Loughran, N., Sampaio, A., & Rashid, A. (2006). From Requirements Documents to Feature Models for Aspect Oriented Product Line Implementation, Berlin, Heidelberg.

Lu, M., & Liang, P. (2017). *Automatic Classification of Non-Functional Requirements from Augmented App User Reviews*. Paper presented at the Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, Karlskrona, Sweden. https://doi.org/10.1145/3084226.3084241

Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., & Brinkkemper, S. (2016). Improving agile requirements: the Quality User Story framework and tool. *Requirements Engineering, 21*(3), 383-403. doi:10.1007/s00766-016-0250-x

Maiden, N., Lockerbie, J., Zachos, K., Bertolino, A., De Angelis, G., & Lonetti, F. (2014). A Requirements-Led Approach for Specifying QoS-Aware Service Choreographies: An Experience Report, Cham.

Mezghani, M., Kang, J., & Sèdes, F. (2018, 20-24 Aug. 2018). *Industrial Requirements Classification for Redundancy and Inconsistency Detection in SEMIOS.* Paper presented at the 2018 IEEE 26th International Requirements Engineering Conference (RE).

Mich, L. (1996). NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA. *Natural Language Engineering, 2*(2), 161-187. doi:10.1017/S1351324996001337

Nguyen, T. H., Grundy, J., & Almorsy, M. (2015). *Rule-based extraction of goal-use case models from text*. Paper presented at the Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, Bergamo, Italy. https://doi.org/10.1145/2786805.2786876

Niemeijer, R. A., de Vries, B., & Beetz, J. (2014). Freedom through constraints: User-oriented architectural design. *Advanced Engineering Informatics, 28*(1), 28-36. doi:10.1016/j.aei.2013.11.003

Ninaus, G., Reinfrank, F., Stettinger, M., & Felfernig, A. (2014, 26-26 Aug. 2014). *Content-based recommendation techniques for requirements engineering.* Paper presented at the 2014 IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering (AIRE).

Ochodek, M., & Nawrocki, J. (2008). Automatic Transactions Identification in Use Cases, Berlin, Heidelberg.

Overmyer, S. P., Benoit, L., & Owen, R. (2001, 19-19 May 2001). *Conceptual modeling through linguistic analysis using LIDA.* Paper presented at the Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001.

Palomba, F., Linares-Vásquez, M., Bavota, G., Oliveto, R., Penta, M. D., Poshyvanyk, D., & Lucia, A. D. (2018). Crowdsourcing user reviews to support the evolution of mobile apps. *Journal of Systems and Software, 137*, 143-162. doi:10.1016/j.jss.2017.11.043

Popescu, D., Rugaber, S., Medvidovic, N., & Berry, D. M. (2008). Reducing Ambiguities in Requirements Specifications Via Automatically Created Object-Oriented Models, Berlin, Heidelberg.

Portugal, R. L. Q., Li, T., Silva, L., Almentero, E., & Leite, J. C. S. d. P. (2018). *NFRfinder: a knowledge based strategy for mining non-functional requirements*. Paper presented at the Proceedings of the XXXII Brazilian Symposium on Software Engineering, Sao Carlos, Brazil. https://doi.org/10.1145/3266237.3266269

Pruski, P., Lohar, S., Aquanette, R., Ott, G., Amornborvornwong, S., Rasin, A., & Cleland-Huang, J. (2014, 25-29 Aug. 2014). *TiQi: Towards natural language trace queries.* Paper presented at the 2014 IEEE 22nd International Requirements Engineering Conference (RE).

Rago, A., Marcos, C., & Diaz-Pace, J. A. (2013). Uncovering quality-attribute concerns in use case specifications via early aspect mining. *Requirements Engineering, 18*(1), 67-84. doi:10.1007/s00766-011-0142-z

Rago, A., Marcos, C., & Diaz-Pace, J. A. (2015, 30 Sept.-2 Oct. 2015). *Identifying duplicate functionality in textual use cases by aligning semantic actions (SoSyM abstract).* Paper presented at the 2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS).

Rago, A., Marcos, C., & Diaz-Pace, J. A. (2016). Assisting requirements analysts to find latent concerns with REAssistant. *Automated Software Engineering, 23*(2), 219-252. doi:10.1007/s10515-014-0156-0

Riaz, M., King, J., Slankas, J., & Williams, L. (2014, 25-29 Aug. 2014). Hidden in plain sight: Automatically identifying security requirements from natural language artifacts. Paper presented at the 2014 IEEE 22nd International Requirements Engineering Conference (RE).

Robeer, M., Lucassen, G., Werf, J. M. E. M. v. d., Dalpiaz, F., & Brinkkemper, S. (2016, 12-16 Sept. 2016). *Automated Extraction of Conceptual Models from User Stories via NLP.*

Paper presented at the 2016 IEEE 24th International Requirements Engineering Conference (RE).

Rodriguez, D. V., Carver, D. L., & Mahmoud, A. (2018, 3-10 March 2018). An efficient wikipedia-based approach for better understanding of natural language text related to user requirements. Paper presented at the 2018 IEEE Aerospace Conference.

Rojas, A. B., & Sliesarieva, E. G. B. (2010). *Automated detection of language issues affecting accuracy, ambiguity and verifiability in software requirements written in natural language.* Paper presented at the Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas.

Rolland, C., & Proix, C. (2013). A Natural Language Approach for Requirements Engineering. In J. Bubenko, J. Krogstie, O. Pastor, B. Pernici, C. Rolland, & A. Sølvberg (Eds.), *Seminal Contributions to Information Systems Engineering: 25 Years of CAiSE* (pp. 35-55). Berlin, Heidelberg: Springer Berlin Heidelberg.

Rooijen, L. v., Bäumer, F. S., Platenius, M. C., Geierhos, M., Hamann, H., & Engels, G. (2017, 4-8 Sept. 2017). *From User Demand to Software Service: Using Machine Learning to Automate the Requirements Specification Process.* Paper presented at the 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW).

Sarmiento, E., Leite, J. C. S. d. P., & Almentero, E. (2014, 26-26 Aug. 2014). *C&L: Generating model based test cases from natural language requirements descriptions.* Paper presented at the 2014 IEEE 1st International Workshop on Requirements Engineering and Testing (RET).

Sateli, B., Angius, E., & Witte, R. (2013, 22-26 July 2013). The ReqWiki Approach for Collaborative Software Requirements Engineering with Integrated Text Analysis Support. Paper presented at the 2013 IEEE 37th Annual Computer Software and Applications Conference.

Sawyer, P., Rayson, P., & Garside, R. (2002). REVERE: Support for Requirements Synthesis from Documents. *Information Systems Frontiers, 4*(3), 343-353. doi:10.1023/A:1019918908208

Schneider, K., Knauss, E., Houmb, S., Islam, S., & Jürjens, J. (2012). Enhancing security requirements engineering by organizational learning. *Requirements Engineering, 17*(1), 35-56. doi:10.1007/s00766-011-0141-0

Silva, B. C. F., Carvalho, G., & Sampaio, A. (2016). Test Case Generation from Natural Language Requirements Using CPN Simulation, Cham.

Slankas, J., & Williams, L. (2013, 8-14 Sept. 2013). *Access Control Policy Extraction from Unconstrained Natural Language Text.* Paper presented at the 2013 International Conference on Social Computing.

Soares, F., Araújo, J., & Wanderley, F. (2015). *VoiceToModel: an approach to generate requirements models from speech recognition mechanisms*. Paper presented at the Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain. https://doi.org/10.1145/2695664.2695724

Sorbo, A. D., Panichella, S., Alexandru, C. V., Shimagaki, J., Visaggio, C. A., Canfora, G., & Gall, H. C. (2016). *What would users change in my app? summarizing app reviews for recommending software changes*. Paper presented at the Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Seattle, WA, USA. https://doi.org/10.1145/2950290.2950299

Stein, C., Etzkorn, L., & Utley, D. (2004). *Computing software metrics from design documents*. Paper presented at the Proceedings of the 42nd annual Southeast regional conference, Huntsville, Alabama. https://doi.org/10.1145/986537.986571

Steinberger, M., Reinhartz-Berger, I., & Tomer, A. (2016). A Tool for Analyzing Variability Based on Functional Requirements and Testing Artifacts, Cham.

Su, X., & Gulla, J. A. (2006). An information retrieval approach to ontology mapping. *Data & Knowledge Engineering, 58*(1), 47-69. doi:10.1016/j.datak.2005.05.012

Thakur, J. S., & Gupta, A. (2016). *Identifying domain elements from textual specifications*. Paper presented at the Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, Singapore, Singapore. https://doi.org/10.1145/2970276.2970323

Thayasivam, U., Verma, K., Kass, A., & Vasquez, R. G. (2011). Automatically Mapping Natural Language Requirements to Domain-Specific Process Models. Paper presented at the Twenty-Third IAAI Conference.

Thomas, K., Bandara, A. K., Price, B. A., & Nuseibeh, B. (2014). *Distilling privacy requirements for mobile applications*. Paper presented at the Proceedings of the 36th International Conference on Software Engineering, Hyderabad, India. https://doi.org/10.1145/2568225.2568240

Thommazo, A. D., Ribeiro, T., Olivatto, G., Werneck, V., & Fabbri, S. (2013, 1-4 Oct. 2013). *An Automatic Approach to Detect Traceability Links Using Fuzzy Logic.* Paper presented at the 2013 27th Brazilian Symposium on Software Engineering.

Tjong, S. F., & Berry, D. M. (2013). The Design of SREE — A Prototype Potential Ambiguity Finder for Requirements Specifications and Lessons Learned, Berlin, Heidelberg.

Wang, C., Pastore, F., & Briand, L. (2018, 9-13 April 2018). *Automated Generation of Constraints from Use Case Specifications to Support System Testing.* Paper presented at the 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST).

Wilson, W. M., Rosenberg, L. H., & Hyatt, L. E. (1997). *Automated analysis of requirement specifications.* Paper presented at the Proceedings of the 19th international conference on Software engineering.

Xiao, X., Paradkar, A., Thummalapenta, S., & Xie, T. (2012). *Automated extraction of security policies from natural-language software documents*. Paper presented at the Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, Cary, North Carolina. https://doi.org/10.1145/2393596.2393608

Yang, H., Willis, A., Roeck, A. D., & Nuseibeh, B. (2010). *Automatic detection of nocuous coordination ambiguities in natural language requirements*. Paper presented at the Proceedings of the IEEE/ACM international conference on Automated software engineering, Antwerp, Belgium. 10.1145/1858996.1859007

Yue, T., Ali, S., & Briand, L. (2011). Automated Transition from Use Cases to UML State Machines to Support State-Based Testing, Berlin, Heidelberg.

Yue, T., Briand, L. C., & Labiche, Y. (2015). aToucan: An Automated Framework to Derive UML Analysis Models from Use Case Models. *ACM Trans. Softw. Eng. Methodol., 24*(3), Article 13. doi:10.1145/2699697

Zeni, N., Kiyavitskaya, N., Mich, L., Cordy, J. R., & Mylopoulos, J. (2015). GaiusT: supporting the extraction of rights and obligations for regulatory compliance. *Requirements Engineering, 20*(1), 1-22. doi:10.1007/s00766-013-0181-8

Zhang, J., & El-Gohary, N. M. (2017). Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Automation in Construction, 73*, 45-57. doi:10.1016/j.autcon.2016.08.027

Zhang, L., Huang, X.-Y., Jiang, J., & Hu, Y.-K. (2017). CSLabel: An Approach for Labelling Mobile App Reviews. *Journal of Computer Science and Technology, 32*(6), 1076-1089. doi:10.1007/s11390-017-1784-1

Zhong, H., Zhang, L., Xie, T., & Mei, H. (2009, 16-20 Nov. 2009). *Inferring Resource Specifications from Natural Language API Documentation.* Paper presented at the 2009 IEEE/ACM International Conference on Automated Software Engineering.

APPENDIX C - SKYZER UAV - LANDING GEAR REQUIREMENTS

Requirements List

1. The landing gear structure shall be designed for a service life equal to that of the Air Vehicle airframe structure.
2. Reversal of the landing gear command during actuation shall result in the landing gear going to the last position commanded.
3. The alternate extension system shall have the capability to extend with the same air speed and maneuver limitations as the retractable landing gear system.
4. An alternate extension system shall be provided For retractable landing gear systems.
5. The time to extend the gear using the alternate extension system shall be _ seconds.
6. The time to extend the gear using the alternate extension system shall commensurate with vehicle operations
7. Failure of door-locking linkages or extension devices shall not prevent alternate unlocking or extension.
8. During any phase of the Air Vehicle Operation a minimum clearance of inches shall be provided between the landing gear and any other part of the aircraft.
9. Clearances between landing gear components shall be such that no unintended contact occurs.
10. Clearances shall apply to grown and spinning tires, landing gear tolerances, and landing gear deflections.
11. A minimum clearance of _ inches shall be provided between landing gear components and stores in their carriage position
12. The minimum clearance shall account for landing gear tolerances and deflections.
13. Retractable landing gears shall have clearance such that with the landing gear in the retracted position and during any transition between the extended and retracted positions there is no contact between the landing gear and any other part of the aircraft
14. The clearance requirement applies to the air loads, accelerations, tolerances and grown and spinning tires for in-transit positions.

15. During all operations, with a flat tire and flat strut and all other conditions nominal, the lowest part of the landing gear structure shall be no closer than 6 inches from the ground and deck obstruction envelopes.
16. During all operations, with a flat tire and flat strut and all other conditions nominal, the door fairing, and Air Vehicle components, including external stores, shall be no closer than 6 inches from the ground and deck obstruction envelopes.
17. During all operations, with a flat tire and flat strut and all other conditions nominal, Air Vehicle components, including external stores shall be no closer than 6 inches from the ground and deck obstruction envelopes.
18. The landing gear shall have natural or augmented damping such that the amplitude of any landing gear oscillations after 3 cycles is reduced to 1/3 and less of an original disturbance, with the exception of brake squeal and chatter.
19. The damping requirement shall apply to all initial displacements of the landing gear at all permissible gross weights.
20. The damping requirement shall apply to all initial displacements of the landing gear at all permissible centers of gravity.
21. The damping requirement shall apply to all initial displacements of the landing gear at all permissible ground speeds on any paved surface with critical components worn to the maximum allowable.
22. The dampening requirement includes shimmy.
23. Joints and wear surfaces shall accommodate repair by providing, at the time of delivery, a minimum of 0.060 inch allowance on the diameter of the lug bore of each pinned joint
24. Joints and wear surfaces shall accommodate repair by providing, at the time of delivery, a minimum of 0.030 inch on each non-circular wear surface.
25. The wear surface requirement shall not apply to parts which will be more economical to replace than repair.
26. No single point non-structural failure, including power interruption or unmated connector, shall prevent gear extension or cause mis-sequencing.
27. The landing configuration shall be defined as the landing gear extended and locked and shock absorber struts fully extended.

28. The landing gear in its landing configuration shall absorb landing energies without producing loads outside its designed envelope for landings at all gross weights and all required runway condition ratings.
29. The Landing energy absorption requirements should reference the UAV weights, load factors and initial landing conditions.
30. The landing gear in its landing configuration (landing gear extended and locked and shock absorber struts fully extended) shall absorb landing energies without producing loads outside its designed envelope for Shipboard landings and bolters at all gross weights.
31. The landing gear shall be able to prevent damage from repeated sudden extension of the shock strut after rebound at landing.
32. The landing gear shall be able to prevent damage from repeated sudden extension of the shock strut from rolling over obstructions.
33. The landing gear shall be able to prevent damage from repeated sudden extension of the shock strut from passing over the deck edge.
34. Landing gear structure and rolling stock shall operate over 1-5/8 inch arresting gear cable and cable supports without damage to the Air Vehicle.
35. No single point non-structural failure shall prevent gear extension.
36. No single point non-structural failure shall cause mis-sequencing.
37. Single point non-structural failure will including power interruption or unmated connector,
38. The landing gear shall retract in _ seconds without damage at all gear airspeed limits and g-loading limits.
39. The landing gear doors shall close in and locking _ seconds without damage at all gear airspeed limits and g-loading limits.
40. For all ground operations, the air vehicle shall maintain operational control and stability such that no part of the air vehicle fuselage or its weapons will contact the ground or permanent ground structures.
41. Permanent ground structures will include servicing equipment, arresting cables, runway lights, etc.
42. The landing gear shall provide for axle or gear jacking at maximum gross weight to permit tire change.
43. The Axle jacking requirement should reference the jacking weights of Air Vehicle
44. The landing gear shall be compatible with CJ67D0250-1 or 31S0A5100-1 axle jacks.

45. The landing gear should be compatible with the MALABAR INTERNATIONAL Model 8732 20 ton axle jack.
46. The landing gear should be compatible with the Model 25 NSN 1730-ND-567-177GxW, CJ67D0250-1, and NSN 1730-00-854-2237RN.
47. The landing gear should be compatible with the CJ67D0250-1 and NSN 1730-00-854-2237RN.
48. Landing gear towing interfaces shall be provided for Air Vehicle towing both forward and aft.
49. Towing shall be compliant with the IAW MIL-STD-805 standard.
50. Landing gear shall contain mooring rings IAW MIL-T-81259(AS) and TD-1B.
51. Mooring rings shall be located such that mooring lines or mooring rings do not interfere with structure, doors, tires, subsystems component tubes and wiring, or other equipment.
52. The landing gear down locks shall remain locked under impact from FOD Up to the maximum gear down speed.
53. Hydraulic and electrical lines shall be located to avoid their failure due to FOD impact.
54. Hydraulic and electrical lines will be integral to the landing gear
55. The landing get shall allow for the shock strut gas charging and inspection for proper servicing without the need for special tools removal of components, or jacking of the complete Air Vehicle.
56. The landing get shall allow for oil filling and inspection for proper servicing without the need for special tools, removal of components, or jacking of the complete Air Vehicle
57. The landing get shall allow for lubrication and inspection for proper servicing without the need for special tools, removal of components, or jacking of the complete Air Vehicle
58. Landing gear shock absorber servicing criteria shall be identical for all shore-based and ship-based operations.
59. The landing gear system shall be capable of performing a maximum sink rate landing at the land-plane landing weight under all operating conditions, within _ seconds after extension and _ seconds after down-lock.
60. When the aircraft is removed from jacks, the shock struts shall stroke smoothly and continuously to the theoretical position for the existing ground load +/- 5 percent of the total stroke.

61. When the aircraft is removed from jacks, the shock struts shall stroke with strut movement that is no greater than _ percent of that strut's 1g static ground load.
62. Landing gear door actuation shall be automatically sequenced with the landing gear actuation.
63. Landing gear door locking shall be automatically sequenced with the landing gear actuation.
64. Retraction of any single landing gear shall not depend on satisfactory operation of any other landing gear.
65. The landing gear shall be restrained in the selected position by operating positive mechanical locks automatically.
66. The down-locks of the landing gear shall not be loaded by ground loads.
67. Up-Locks and down-locks shall be designed such that rigging requires only simple adjustment and does not require devices with close tolerance adjustments.
68. Ground lock pins shall be provided.
69. Ground lock pins shall prevent landing gear retraction by any means without damage to the gear or ground lock pins.
70. ground lock pins shall allow for removal and insertion with the aircraft unpowered.
71. The centers of wheel axles shall have at least 6.5 inches of clearance with flat tires.
72. To reduce tire rollover and scrubbing, tire inboard or outboard movement tires shall not occur during strut compression and strut extension.
73. Six inches of ground clearance shall be maintained for all configurations of the aircraft.
74. The wheel shall be retained on the axle in case of bearing failure.
75. Hydraulic components within the Landing Gear System shall conform to SAE AS5440.
76. Hydraulic components within the Landing Gear System shall conform to SAE AS8775.
77. The Landing Gear shall be designed to prevent fluids from pooling on their surfaces when extended.
78. The Arresting Hook Systems shall be designed to prevent fluids from pooling on their surfaces when extended.
79. For multi-wheeled struts, upon one tire failure at any time throughout a mission, the remaining tires shall survive the remainder of the mission.

VITA

Education:

Master of Science - **Technical University of Munich, Germany** Oct 2018

Bachelor of Science - **Technical University of Munich, Germany** Aug 2015

Professional Experience:

NASA Jet Propulsion Laboratory - Software Systems Engineer since Feb 2023

Systems Engineering Research Center (SERC) - Researcher Jan 2022 - Jan 2023

Knorr-Bremse SfS - Junior Mechanical Engineer Jun 2014 - Apr 2019

New York Air Brake - Intern for Product Development & Design Apr 2017 - Sep 2017

Publications:

D. Dunbar, M. Vierlboeck, & M. Blackburn, "Use of Natural Language Processing in Digital Engineering Context to Aid Tagging of Model", **2023 Annual IEEE International Systems Conference (SysCon)**, 2023.

L. Xiao, G. Zhao, M. Vierlboeck, & R. R. Nilchiani, "A Framework on Early Decoupling Level Metric Assessment based on NLP4RE", **2023 International Conference on Systems Engineering Research (CSER)**, 2023.

M. Vierlboeck, R. R. Nilchiani, A. Ganguly, & C. M. Edwards, "Requirements Engineering for the Development of Disruptive Systems Engineering Innovations," 2**022 IEEE International Symposium on Systems Engineering (ISSE)**, 2022, DOI: 10.1109/ISSE54508.2022.10005334.

C. M. Edwards, R. R. Nilchiani, A. Ganguly, & M. Vierlboeck, "Evaluating the Tipping Point of a Complex System: The Case of Disruptive Technology," **TechRxiv**, 2022, DOI: 10.36227/techrxiv.21126292.v1.

M. Vierlboeck, R. R. Nilchiani, M. Blackburn, "Natural Language Processing to Assess Structure and Complexity of System Requirements," **TechRxiv**, 2022, DOI: 10.36227/techrxiv.20818984.v1.

M. Vierlboeck, D. Dunbar, & R. R. Nilchiani, "Natural Language Processing to Extract Contextual Structure from Requirements", **2022 Annual IEEE International Systems Conference (SysCon)**, 2022, DOI: 10.1109/SysCon53536.2022.9773855.

M. Vierlboeck, C. Lipizzi, & R. R. Nilchiani, "Natural Language in Requirements Engineering for Structure Inference – An Integrative Review," **arXiv**, 2022, DOI: 10.48550/arXiv.2202.05065.

M. Vierlboeck & R. R. Nilchiani, "Complexity at the Requirement Stage – A Novel Paradigm for System Development," **2021 SERC Doctoral Student Forum**, 2021.

M. Vierlboeck & R. R. Nilchiani, "Requirement Engineering in the Age of System and Product Complexity – A Literature Review," **2021 IEEE International Symposium on Systems Engineering (ISSE)**, 2021, pp. 1-8, DOI: 10.1109/ISSE51541.2021.9582439.

M. Vierlboeck, R. R. Nilchiani, & C. M. Edwards, "The Pandemic Holiday Blip in New York City," in **IEEE Transactions on Computational Social Systems**, vol. 8, no. 3, pp. 568-577, June 2021, DOI: 10.1109/TCSS.2021.3058633.

M. Vierlboeck & R. R. Nilchiani, "Effects of COVID-19 Vaccine Developments and Rollout on the Capital Market – A Case Study," **arXiv**, 2021, DOI: 10.48550/arXiv.2105.12267.

A. Pugliese, R. R. Nilchiani, & M. Vierlboeck, "Structural Complexity Analysis to Evaluate Technical Risk in Defense Acquisition," **Acquisition Research Program**, 2021, NPS handle: 12345678/4378.

M. Vierlboeck, R. R. Nilchiani, & C. M. Edwards, "Systems Approach to Localize Tipping Points for the Emergency Services in Face of the COVID-19 Pandemic," **2020 IEEE International Symposium on Systems Engineering (ISSE)**, 2020, pp. 1-9, DOI: 10.1109/ISSE49799.2020.9272229.

M. Vierlboeck, R. R. Nilchiani, & C. M. Edwards, "The Easter and Passover Blip in New York City: How exceptions can cause detrimental effects in pandemic times," **medRxiv**, 2020, DOI: 10.1101/2020.04.14.20065300.

M. Vierlboeck, K. Gövert, J. Trauer, and U. Lindemann, "Budgeting for Agile Product Development," **2019 International Conference in Engineering Design (ICED)**, 2019, DOI: 10.1017/dsi.2019.223.

Awards and Honors:

2023 Fabrycky-Blanchard Award

2022 Excellence Doctoral Fellowship Recipient (awarded to the best 6 Ph.D. students)

2022 Third Place in Heat 2 of the AIRC & DLA Defense Data Grand Prix

2022 Distinguished Teaching Assistant Award Winner

2022 Best School of Systems and Enterprises Student Paper Award Winner

2021 Lockheed Martin Fellowship Recipient

2019 Stevens Institute of Technology Provost's Doctoral Fellowship Recipient