

12-2009

# Linguistic Analysis of Natural Language Engineering Requirements

Carl Lamar

Clemson University, [clamarcu@gmail.com](mailto:clamarcu@gmail.com)

Follow this and additional works at: [https://tigerprints.clemson.edu/all\\_theses](https://tigerprints.clemson.edu/all_theses)

 Part of the [Engineering Mechanics Commons](#)

---

## Recommended Citation

Lamar, Carl, "Linguistic Analysis of Natural Language Engineering Requirements" (2009). *All Theses*. 671.  
[https://tigerprints.clemson.edu/all\\_theses/671](https://tigerprints.clemson.edu/all_theses/671)

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact [kokeefe@clemson.edu](mailto:kokeefe@clemson.edu).

LINGUISTIC ANALYSIS OF NATURAL LANGUAGE  
ENGINEERING REQUIREMENTS

---

A Thesis  
Presented to  
the Graduate School of  
Clemson University

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
Mechanical Engineering

---

by  
Carl Lamar  
December 2009

---

Accepted by:  
Dr. Gregory M. Mocko, Committee Chair  
Dr. Joshua D. Summers  
Dr. Paul Venhovens

## ABSTRACT

In engineering design, the needs of the customer are expressed through engineering requirement statements. These requirement statements are often expressed using natural language because they are easily created and read. However, there are several problems associated with natural language requirements including but not limited to ambiguity, incompleteness, understandability, testability and over specificity. Several representation and analysis tools have been proposed to address these problems within a requirement statement. These tools include formal languages, such as UML and SysML, requirement management tools, such as IBM Telelogic Doors, and natural language processors such as QuARS. These tools assist in the systematic elicitation and creation of requirements, improve requirement visibility and traceability, and provide a central repository for shared access. However, these tools do not prescribe a formal representation of a requirement and its elements. The effectiveness of these tools can be greatly improved with a formalized syntax for expressing engineering requirements.

The research presented in this thesis examines engineering requirements from a linguistic viewpoint and leads to a formalized syntax based on parts of speech, grammatical functions, and sentence structure. Specifically, a requirement statement is decomposed into four syntactical elements: artifact, necessity, function, and condition. Further, grammar and linguistics provide the basis for requirements classification into functional or non-functional and qualitative or quantitative requirements. Finally, the deficiencies in current natural language requirements such as incompleteness,

understandability, ambiguity, and specificity, are identified through the formal syntax and grammatical rules. The requirements syntax and analysis method are demonstrated on 110 requirements from the Family of Medium Tactical Vehicles (FMTV). Using the syntax and analysis method proposed, the count of incomplete requirements, percentages of function and non-functional requirements, and specificity of the requirement statements in the document were determined. Identifying such requirement measures will help to improve the expressiveness of requirement statements and help to identify if appropriate requirements are being authored for the different stages of design (i.e. conceptual, embodiment, detailed). To further improve the analysis method proposed, more quality attributes of requirement statements have to be addressed such as ambiguity and traceability. The end goal is to develop a syntax and analysis method that addresses all quality attributes of a requirement statement that is not empirically based but rule based.

## DEDICATION

I would like to dedicate this thesis to my parents, family, and friends who supported me throughout the writing and research process.

## ACKNOWLEDGMENTS

I would like to acknowledge the professors I worked most closely with, Dr. Mocko, Dr. Summers, Dr. Venhovens and Dr. Fadel for supporting me and for allowing me to pursue my research ideas in engineering design. I would like to acknowledge Clemson's mechanical engineering staff, especially Ms. Tameka Boyce and Ms. Gwen Dockins, for assisting me with the technicalities of being a graduate student. Finally, I would like to acknowledge my lab mates, C. Sen, B. Caldwell, and D. Norfleet for adding to my personal growth as a graduate student at Clemson.

## TABLE OF CONTENTS

Abstract.....	i
Dedication.....	iii
Acknowledgments.....	iv
Table of Contents.....	v
List of Tables.....	vii
List of Figures.....	viii
Chapter One: motivation and problem statement.....	1
Chapter Two: Literature Review of current methods to analyze and express requirement statements.....	7
2.2 Requirements Further Explained.....	17
2.3 Research Questions.....	29
Chapter Three: Fundamental Basis for expressing and analyzing engineering requirements.....	31
3.2 Grammar and linguistics directly applied to requirement statements.....	44
Chapter Four: Requirement Syntax and analysis method.....	51
4.1 Syntax.....	51
4.2 Linguistic Analysis Process.....	55
4.3 Quality Attributes Addressed by Analysis Method.....	59
Chapter Five: FMTV Test case.....	63
Chapter Six: Conclusion and future work.....	76
6.1 Answers to Research Questions.....	76
6.2 Contribution.....	79
6.3 Future Work.....	81
Appendices.....	83

Appendix A: Penntree Bank Tag Set [26] .....	84
References.....	85



## LIST OF TABLES

Table 2.1	Quality indicators mapped to quality attributes [10] .....	8
Table 2.2	Quality properties and indicators from the Linguistic Approach to the Natural Language Requirements Quality (Fabbrini, et al. 2001).....	11
Table 2.3	Abilities of engineering requirements [Hull].....	14
Table 2.4	Illustrates how requirements can measured [7] .....	21
Table 2.5	Quality attributes from SATC [10].....	24
Table 2.6	Quality indicators from SATC [10].....	25
Table 2.7	Examples for the imperatives used by SATC [10] .....	26
Table 3.1	Eight parts of speech of traditional grammar [20].....	33
Table 3.2	Examples of inconsistencies within tagged sentences .....	35
Table 3.3	Grammatical functions [23].....	36
Table 3.4	Common Modal Verbs.....	43
Table 4.1	Grammatical and linguistic elements that are used as the foundation of the NL requirement syntax.....	51
Table 4.2	BNF general notation.....	53
Table 4.3	BNF of general requirement .....	54
Table 4.4	BNF of functional requirement.....	55
Table 4.5	BNF of non-functional requirement .....	55
Table 5.1	Sample set of FMTV requirements.....	64
Table 5.2	FMTV requirements with missing subject.....	68
Table 5.3	Non-functional requirements where the subject complement is a noun .....	69
Table 5.4	Non-functional requirements that consist of participle.....	70
Table 5.5	Non-functional requirement where the subject complement is an adjective ...	70
Table 5.6	Examples of requirements with no numerical values .....	73

## LIST OF FIGURES

Figure 1.1	Flow chart of engineering design process [3].....	2
Figure 2.1	Quality indicators for expressiveness component of quality model [11] .....	22
Figure 3.1	Four tiered approach to linguistic analysis of engineering requirements.....	32
Figure 3.2	A tagged requirement statement using the Penn Treebank tag set.....	34
Figure 3.3	Subject and predicate.....	40
Figure 3.4	Mapping of NL requirement elements to NL sentence elements .....	44
Figure 3.5	Subject organization within a requirement [23] .....	45
Figure 4.1	Flow chart describing linguistic analysis process .....	56
Figure 5.1	Snippet from FMTV requirements analysis spreadsheet.....	66
Figure 5.2	Completeness of FMTV requirements .....	67
Figure 5.3	Main verb types .....	68
Figure 5.4	Breakdown of parts of speech following the linking verb .....	71
Figure 5.5	Count of numerical values present in the examined requirements.....	72
Figure 5.6	Count of adjuncts of examined requirement statements.....	73
Figure 5.7	Adjuncts compared to numerical values in examined requirements .....	74

## CHAPTER ONE: MOTIVATION AND PROBLEM STATEMENT

Engineering requirements describe functions or characteristics that must be fulfilled by a product. Requirements express the needs of several stakeholders including multi-disciplinary engineering designers, software developers, manufacturing engineers, industrial designers, end users, marketing and sales, and maintenance personnel. Further, requirements define an expectation of the design solution and constrain the solution space of the solutions [1]. Thus, it is important to ensure that the stakeholders in the design process are generating solutions, developing simulations, and verifying concepts for a consistent set of requirements.

Producing correct engineering requirements is essential in producing design solutions that satisfy the end user. The development of requirement documents is one of the first tasks undertaken when designing a product. From Figure 1.1 it can be seen that engineering requirements begin early in the design process and they are carried throughout the entire design process, getting further refined along the way. The requirements developed at the beginning of the design process will affect the conceptual, embodiment, and detail design phases [2]. Thus, it is important to ensure that the stakeholders in the design process are generating solutions, developing simulations, and verifying concepts for a consistent set of requirements.

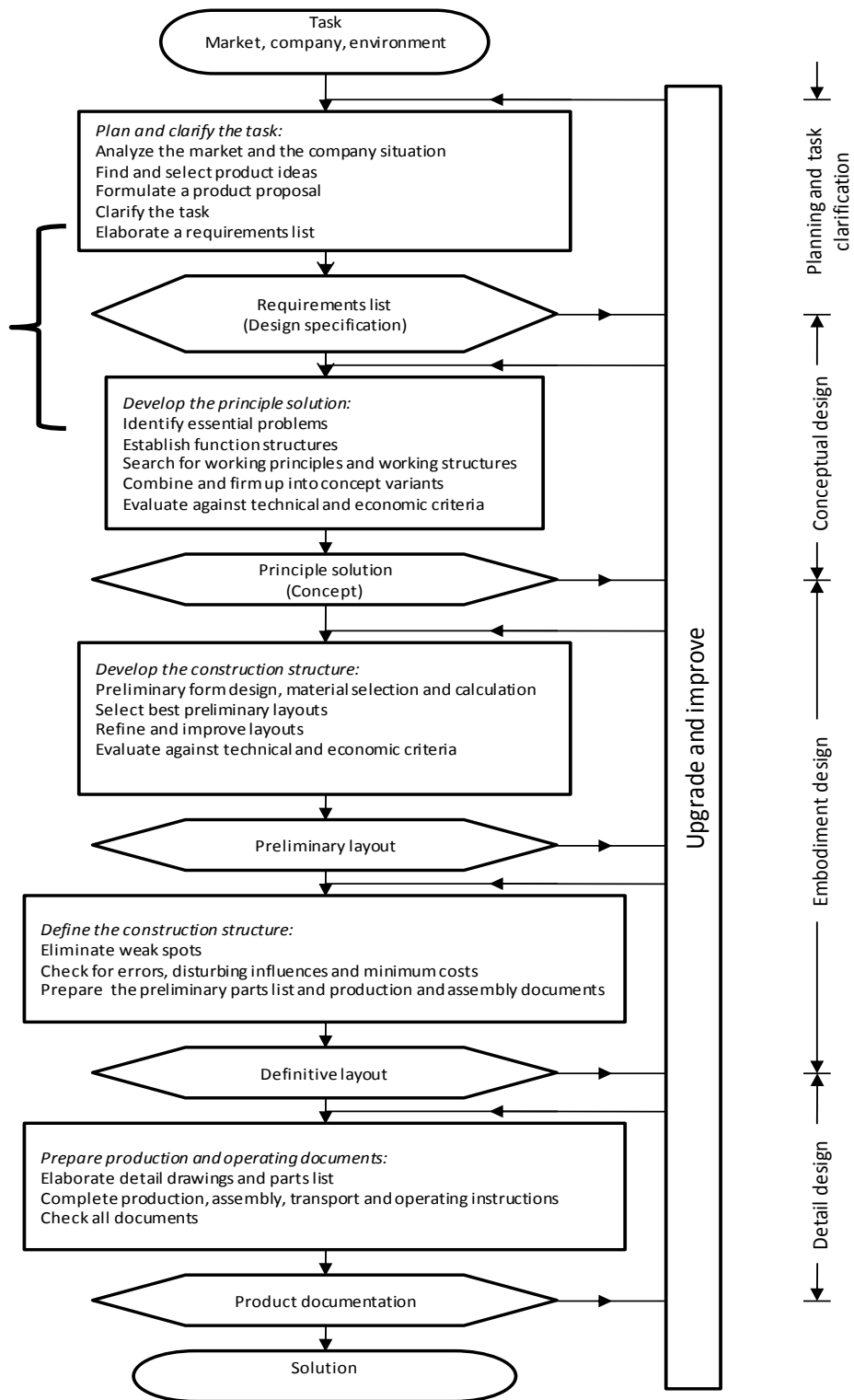


Figure 1.1 Flow chart of engineering design process [3]

Most often engineering requirements start out as natural language sentences that follow the same grammatical rules as any other type of English sentence. Requirement statements are articulated using the words and symbols and adhere to grammatical rules from a chosen language [1]. Natural language (NL) is used to express and document engineering requirements using document-based approaches because it is often the spoken language of the designer and lend to the easy documentation of engineering requirements. An engineering requirement statement may be supplemented, or further clarified, using graphical or supporting documents. This is supported in requirements management tools such as IBM Rational DOORS and formal languages such as SysML. However, the natural language requirement statement is the crux of requirements documents.

There are several problems associated with natural language requirements in the context of engineering design, particularly in computer-supported product development. The problems associated with NL requirements include but not limited to (1) ambiguity of requirements between customers [4], (2) incompleteness of requirement statements [5] [6], and (3) over specificity [7] . Further, Grady [1] identifies three key issues in formulating requirements as (i) problems associated with expressing requirements in the chosen language, (ii) technical knowledge deficiencies to understand the underlying requirements, and (iii) difficulty in specifying what the requirement describes. Requirements stated in this manner typically lack consistency in expressiveness which makes it difficult to analyze or process a set of requirements. These issues are tightly inter-related, often resulting in poor quality requirements.

As stated previously, formal languages such as UML and SysML have been developed that supplement text based requirement documents by introducing graphical relations between requirements to show hierarchal relationships, derived relationships, relationships showing refinement, and relationships showing requirements have been verified [8]. . These formal languages allow requirement developers to better exploit requirement statements but they only take into account the requirement as a single text-based entity. . These text-based requirements (TBRs) are text strings that represent a single engineering requirement statement, thus limiting the ability to reason and query based on the components of a requirements sentence. Current research and development efforts include the International Council on Systems Engineering (INCOSE), the development of the Systems Modeling Language (SysML) [8], and current efforts at NIST on the development of ISO 10303 (STEP standards). Specifically, STEP application protocol AP233: Systems Engineering Data Exchange standard is being developed to enable system engineering activities. These languages and tools are limited to TBRs, providing basic modeling for tracing requirements, linking requirements, classifying requirements, decomposing requirements, assigning requirements to physical systems, and including supplementary information.

To reduce the problems associated with NL requirements expressiveness, standardized boilerplates and templates have been developed. For example, Hull and colleagues [6] propose a structure for specifying and writing requirements. These boilerplates allow for global changes in style to be effected, system information can be processed more easily, and confidential information can be protected [6]. Further, MIL-

STD-961D/E [9] provides best practices for writing requirements and specifications. The standard provides a reduced vocabulary for writing requirements and rules for specifying what is included in a requirement. An issue with standardized templates is the difficulty associated with enumerating all possible templates [1]. The templates provide a means for writing requirements as well as classifying the different types of requirements including: performance, interoperability, sustainability, and environmental. Thus, if all requirements cannot be expressed in templates, it may be difficult to perform advances reasoning. Second, boilerplates are based on domain and knowledge specific terminology that is at a much higher level than natural language, thus limiting the benefits of natural language and linguistic processing. Standardized templates are often based on a finite set of pragmatic terms that is designer or domain specific, and thus limits the standardization of a requirements representation. While customized and reduced vocabulary sets are important for specific requirements, there is a need to establish a general standard for engineering requirements that spans across discipline in the product lifecycle that is based on basic elements of language. This would eliminate the creation of new vocabulary set when new products are developed. Finally, existing requirements templates are limited in computational representation. For example, computational representations and standardized models have not been proposed in military standards (MIL-STDs), thus limiting the ability to store and exchange requirements across a wide range of disparate stakeholders in the development process.

To even better capture and analyze requirements, this research will focus on both developing a method to better express and analyze NL engineering requirements and also

presenting a method to analyze the individual constituents, rather than a single entity, that make up a requirement statement

Many of the current guidelines and tools use a pragmatic approach to address the issues concerning engineering requirements. This type of approach only captures certain aspects of a requirement. In the research presented in this thesis, a linguistic approach will be taken to address the underlying issues with associated with documenting and analyzing engineering requirements. This linguistic approach will provide a theoretical basis for characterizing and modeling technical requirements. The key contributions in this research are twofold. First, a formalized syntax that will guide users to create more complete, understandable, and unambiguous requirements, and second a method to analyze engineering requirements will be presented.



## CHAPTER TWO: LITERATURE REVIEW OF CURRENT METHODS TO ANALYZE AND EXPRESS REQUIREMENT STATEMENTS

A literature review was conducted on methods used to express and analyze engineering requirements. These methods included a review of quality models used to grade engineering requirements, a review of guidelines used to express natural language engineering requirements, and methods to analyze these requirements.

### 2.1.1 Models for evaluating the quality of engineering requirements

Wilson and co-authors [10] identify nine metrics for evaluating the quality of NL requirements for software design based on the frequency of word or phrases used. These quality metrics serve as a basis for understanding what quality attributes a NL requirement should possess and how they can be used to improve the quality of requirements. The first quality model examined was the Automated Quality Analysis of Natural Language Requirement Specification tool was developed by Software Assurance Technology Center (SATC) to objectively quantify the quality of a requirements document [10]. This was achieved by compiling a single list of desirable quality attributes that requirement documents should exhibit.

**Table 2.1 Quality indicators mapped to quality attributes [10]**

Quality Indicators (measurables)	Quality Attributes										
	Complete	Consistent	Correct	Modifiable	Ranked	Testable	Traceable	Unambiguous	Understandable	Validatable	Verifiable
Imperatives	x			x			x	x	x	x	x
Continuances	x			x	x	x	x	x	x	x	x
Directives	x		x			x		x	x	x	x
Options	x					x		x	x	x	
Weak Phrases	x		x			x		x	x	x	x
Size	x					x		x	x	x	x
Text Structure	x	x		x	x		x		x		x
Specification Depth	x	x		x			x		x		x
Readability				x		x	x	x	x	x	x

In order to use these to determine the quality of a requirements document, quantitative attributes must be developed that relate back to the quality attributes. Nine quality indicators were developed by SATC based on a set of NASA requirement documents. These indicators are based on frequently used words or phrases. The quality indicators are then aggregated to quality attributed. The indicators are broken into two categories: indicators relating to individual requirements and indicators relating to the entire requirements document [10].

- Quality Indicators of individual requirements [10]
  - Imperatives (modal) – Phrases that command that something must be provided (i.e., shall and must)

- Continuances – Phrases that indicates the organization and structure of the requirements. Continuances follow the imperative. (i.e., below, as follows, in particular)
- Directives – Phrases that point to illustrate information within the document. Strengthens the documents specification statements and makes more understandable. (i.e. figure, table, for example)
- Options – Words that give the developer latitude in satisfying the specification statements. These words loosen the specification, reduces acquirers control over final product. (i.e. can, may, optionally)
- Weak Phrases – Phrases that are apt to cause uncertainty. (i.e. adequate, as a minimum, be able to)
- Quality Indicators for entire document (Objective):
  - Size – Total number of:
    - Lines of text
    - Imperatives
    - Subjects of specification statements
    - Paragraphs
  - Specification Depth – Used to report number of imperatives found at each of the documents levels. This reflects the structure of requirement statements and helps to indicate how concise the document is.
  - Readability – How easily requirements are read and understood

- Text Structure – Report number statement identifiers found at each hierarchical level. Indication of documents organization consistency and level of detail. Most detailed documents typically has 9 levels.

Another research effort started at Software Engineering Institute at Carnegie Mellon University was focused on processing natural language requirements through a derived quality model [7]. Cabrini and colleagues [11] present a method for the analysis of natural language requirements based on a derived quality model. The quality model is composed of quality properties that requirements should exhibit and aims at addressing issues with requirements without increasing the formalism level. The quality model is separated into four high-level quality properties of NL requirements. The four quality properties are non-ambiguity, specification completion, consistency, and understandability. Non-ambiguity is the capability for each requirement to have a unique interpretation. Specification completion is the capability of each requirement to uniquely identify its object or subject. Consistency addresses the requirements capability to avoid potential or actual discrepancies. Understandability represents the capability of a requirement to be fully understood [11]. Similar to the tool developed by SATC, these quality properties are mainly subjective and have to be evaluated using quantitative quality indicators. These quality indicators affect both individual requirements and the entire document. The quality indicators are identified by keywords that have been defined from the analysis of several requirement documents. The table below shows the quality properties, the quality indicators, and whether it affects individual requirements or whole requirement documents.

**Table 2.2 Quality properties and indicators from the Linguistic Approach to the Natural Language Requirements Quality (Fabbrini, et al. 2001)**

<b>Quality Properties</b>	<b>Quality Indicator</b>	<b>Individual Requirements</b>	<b>Whole Requirement Documents</b>
<b>Non - Ambiguity</b>	Vagueness	x	
	Subjectivity	x	
	Optionality	x	
	Weakness	x	
<b>Specification Completion</b>	Under-specification	x	
<b>Consistency</b>	Under-reference		x
<b>Understandability</b>	Multiplicity	x	
	Implicity	x	
	Unexplanation		x

### 2.1.2 Guidelines and Best Practices for Expressing and Writing NL requirements

Hook developed guidelines for expressing requirements and suggests best practices for writing good requirements [5]. Furthermore, common problems and pitfalls in requirements documentation are identified and strategies for avoiding them are presented. For example, the guidelines state that good requirements should be necessary, verifiable, attainable, and should express a single thought. The most common problems observed in writing requirements are [5]:

- Making bad assumptions
  - Occur because designer does not have a sufficient amount of information
- Writing “how” instead of “why”
- Using incorrect terms

- Using incorrect sentence structure or bad grammar
- Missing requirements
- Over-specifying

Some of these issues are related to issues and limited design knowledge, while others are related to ambiguity within requirements statements. The former cannot be completely eliminated through formal methods and quality metrics. However, it is conjectured that ambiguous requirements will be reduced. Guidelines are presented that prescribe the use of terms, structure, and grammar that assists in avoiding the aforementioned common problems. The guidelines concerning use of terms state that requirement authors should understand the use of shall, will, and should and maintain consistency of their usage throughout the document. Term guidelines are also presented that detail what terms should not be used within a requirement sentence because they create ambiguous or unverifiable requirements. Examples of these terms are stated below:

- Support
- But not limited to
- Etc.
- And/or
- Minimize
- Maximize
- Rapid
- User-friendly
- Easy

- Sufficient
- Adequate
- Quick

The guidelines fall short of providing any linguistic analysis of the terms that should not be used. Linguistic analysis of the above words will show relationship between these words and the other parts of an NL requirement which will better present why those words should not be used. One of these methods is a prescribed structure and grammar.

A structure is also presented that demonstrates how natural language requirements should be expressed.

- *The system shall provide....*
- *The system shall be capable of....*
- *The system shall weigh...*
- *The subsystem #1 shall provide....*
- *The subsystem #2 shall interface with....*

The sentence structure is loosely defined and does not detail how an entire requirement should be structured or how the elements are related to one another. Further, in a comprehensive book written by Hull *et al* [6] methods and guidelines are presented for the purpose of specifying how engineering requirements should be written and documented. The authors identify several key abilities of requirements, summarized in Table 2.3

**Table 2.3 Abilities of engineering requirements [Hull]**

- Ability to uniquely identify every statement of a requirement
- Ability to classify every statement of requirement in multiple ways, such as: by importance, type, urgency
- Ability to track the status of every statement of requirement, in support of multiple processes, such as: review status, satisfaction status, qualification status
- Ability to elaborate a requirement in multiple ways, such as by providing: performance information, quantification, test criteria, rationale, comments
- Ability to view a statement of requirement in the document context, i.e. alongside its surrounding statements
- Ability to navigate through a requirements document to find requirements according to a particular classification or context
- Ability to trace to any individual statement of requirement

In addition to detailing these abilities, several boilerplates for expressing requirements are proposed. These boilerplates are templates that break down a requirement into several main parts (i.e. system, function, object, performance, units). An example of the performance boilerplate is shown below.

*The <system> shall <function><object> every <performance><units>*

There are several boilerplates that are tailored to the different requirement types such as performance, interoperability, sustainability, and environmental. These boilerplates allow for global changes in style to be effected, system information can be processed more easily, and confidential information can be protected [6]. However, the author does not fully detail what types terms should be inserted into the template to ensure a correct requirement of that type. Further, MIL-STD-961D/E [9] provides best practices for writing requirements and specifications. The standard provides a reduced vocabulary for



writing requirements and rules for specifying what is included in a requirement. An issue with standardized boilerplates and templates is the difficulty associated with enumerating all possible templates [1]. Thus, if all requirements cannot be expressed in templates, it may be difficult to perform advances reasoning. Second, boilerplates are based on domain and knowledge specific terminology that is at a much higher level than natural language, thus limiting the benefits of natural language and linguistic processing. Standardized templates often use language that is designer or domain specific and thus limits the standardization of a requirements representation. While customized and reduced vocabulary sets are important for specific requirements, there is a need to establish a general standard for engineering requirements that spans across discipline in the product lifecycle that is based on basic elements of language. Finally, existing requirements templates are limited in computational representation. For example computational representations and standardized models have not been proposed in military standards (MIL-STDs), thus limiting the ability to store and exchange requirements across a wide range of disparate stakeholders in the development process.

### 2.1.3 Analyzing NL requirements

#### *2.1.3.1 Processing Natural Language Software Requirement Specifications [12]*

Osborne *et al* [12] discuss how natural language techniques can be used to detect and resolve ambiguities in requirement documents. There are four major elements of a natural language processor: grammar component, lexicon component, semantic component, and a parser. The grammar handles the syntax of the requirement, the

lexicon deals with the meanings of the words, and the semantics handles the meanings of the parsed sentences. The parser is the program that analyzes a sentence and produces the phrase structure trees to be analyzed by the linguistic components. From here problems with using an unrestricted natural language with a NLP tool are discussed. The problems are:

- Lexicon may fail to contain entries for all the words that the system might encounter
- Grammar might assign more than one parse to a single sentence
- Semantics might fail to account for all constructs of a sentence that the system parsed

These problems are addressed by proposing a controlled natural language to be used because of its limited scope. However, the use of a controlled language does allow for some inherent problems which are described below:

- Reduces the habitability of the system (too restricting)
- User needs guidance on how to phrase requirements in terms of the CL
- NL are not always appropriate medium for expressing all requirements (i.e. algorithms)

To address these problems, emphasis is placed on choosing an appropriate controlled language. The Alvey Natural Language Toolkit is the controlled natural language, because of its broad covering of grammars, a lexicon of 40,000 entries, and semantic component that assigns one or more logical forms to each parsed sentence [12]. Several

additions are amended to the controlled language, such as a tool to provide feedback to the user, a parse selection mechanism to rank multiple parses, error diagnoses to handle unparsable sentences, and a means to present the ambiguous requirements.

The problems with the controlled natural language processor that are pertinent to this thesis are lack of a pragmatic component to ensure style guidelines, and the difficulties in determining how parses differ. A rule based formalized syntax in conjunction with the natural language processor will be more apt of correctly addressing ambiguity in requirement specifications.

## **2.2 Requirements Further Explained**

Engineering requirements describes either how a system behaves or properties of the system. According to IEEE [13] a requirement is:

- (1) A condition or capability needed by a user to solve a problem or achieve an objective.
- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
- (3) A documented representation of a condition or capability as in (1) or (2) [13].

At the highest level requirements can be functional or non-functional and qualitative or quantitative. Functional requirements are requirements that specify a function that a system or system component must be able to perform [13]. Non functional requirements tend to describe properties of a system [14]. Functional and non-functional requirements

can also be qualitative or quantitative. Categorizing requirements as qualitative or quantitative helps to identify how specific or vague the requirement is. Ideally as the requirement process is advanced, requirements should start with qualitative non-functional requirements, which define the goals of the design [15], and the ending requirements document should detail the form of the design. These are typically non-functional quantitative requirements (form requirements).

### 2.2.1 Current Guidelines for Requirements

The successfulness of an engineering design is not simply based on just the existence of requirements; the requirements must be expressed correctly. Correctly expressed requirements are verifiable, attainable, clear, and state something that is necessary to the design [5].

A requirement is verifiable if a process or test exists that can verify if the system being designed meets that requirement [16]. Attainable requirements can be achieved with readily available resources such as budget, knowledge, and time. Clear requirements are able to express the essential statement in a manner where it is easily understood and concise [5]. These aforementioned attributes apply to individual requirements, but similar attributes also apply to the entire requirements document. A good requirements document should be unambiguous, complete, verifiable, consistent, modifiable, traceable, and usable during maintenance and operation of the system [16]. Although these attributes pertain to an entire requirement documents they may also apply to the individual requirements. For example, if a requirement contains one ambiguous requirement then the entire documents is considered ambiguous.

All engineering requirements are derived from customer needs [17]. This transformation of customer needs to engineering requirements inherently leads to incorrect requirements. These incorrect requirements originate from misinterpretations poorly written requirements, or poorly expressed requirements. How well these customer needs are transformed into engineering requirements often determines the success of a design. There has been research in the area of requirements elicitation, however a single agreed upon systematic process for requirements elicitation does not exist. Instead guidelines are presented to ensure that requirements are indeed correct. Pahl and Beitz's elicitation guidelines, focuses on concentrating a requirements document down to its essential statements. The steps are presented below

Pahl and Beitz [3]

- Eliminate personal preferences.
- Omit requirements that have no direct bearing on the function and the essential constraints.
- Transform quantitative into qualitative data and reduce them to essential statements.
- As far as it is purposeful, generalize the results of the previous step.
- Formulate the problem in solution-neutral terms.

Ullrich and Eppinger also have guidelines regarding the elicitation of engineering requirements. Their guidelines focus more on transforming customer needs into engineering requirements. Their guidelines are presented below.

Ullrich and Eppinger [17]

- Express the need in terms of what the product has to do not in terms of how it might do it.
- Express the need as specifically as the raw data.
- Use positive, not negative phrasing.
- Express the need as an attribute of the product.
- Avoid the words must and should.

The guidelines presented by the authors above mainly focused on elicitation techniques and transformation techniques that create engineering requirements. However, these guidelines do not focus on how to express engineering requirements. Ivy Hooks has guidelines that detail how to write good requirements. In the report, Hooks [5] addresses common problems in writing good requirements which are listed below:

- Making bad assumptions
- Writing implementation (how) instead of (what)
- Describing operations instead of writing requirements
- Using incorrect terms
- Using incorrect sentence structure or bad grammar
- Missing requirements
- Over-specifying

## 2.2.2 Representing Natural Language Requirements

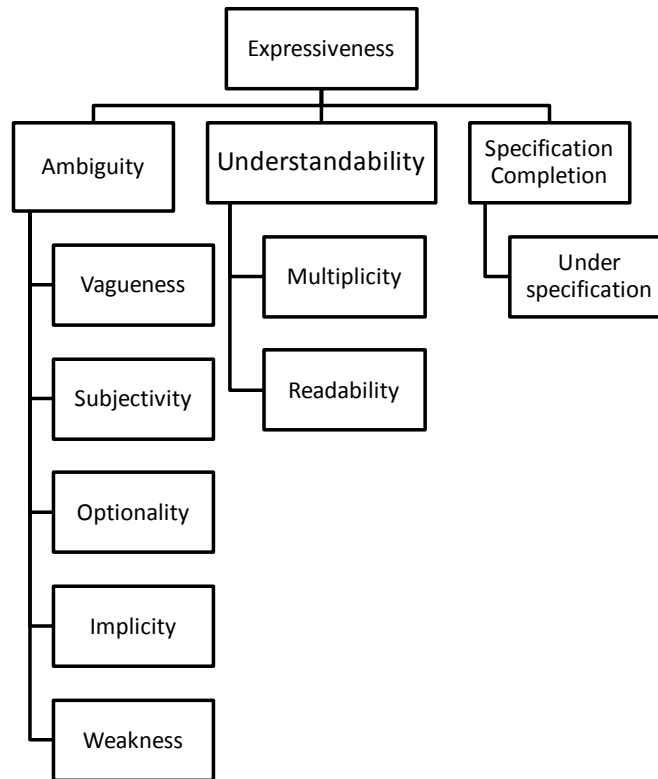
Natural language requirements are expressed using human written language which is not considered a formal representation. Requirement documents using natural language requirements often are created from various sources within an organization such as marketing or engineering. The level of correctness of the requirement documents is in large part determined by the linguistic capabilities of the different sources [11]. Based on the model presented by [7], the correctness of natural language requirements can be measured based on three quality properties, expressiveness, consistency, and completeness from the lexical, syntactic, and semantic view points [7].

**Table 2.4 Illustrates how requirements can measured [7]**

		Lexical	Syntactic	Semantic
Expressiveness				
	Unambiguity			
	Understandability			
	Specification Completion			
Consistency				
Completeness				

Each of the three subjective constituents associated with expressiveness; ambiguity, understandability, and specification completion, has objective quality indicators associated with them which is shown in Figure 2.1. For ambiguity the indicators are vagueness, subjectivity, optionality, implicitness, and weakness. For

understandability the quality indicators are multiplicity and readability. For specification completion, under-specification is the only indicator.



**Figure 2.1 Quality indicators for expressiveness component of quality model [11]**

A full analysis of a natural language requirement will consist of examining a requirement using the aforementioned measures from the syntactic, semantic and lexical viewpoints. The syntactical point of view examines the three measures based on how the words of the requirements are put together. Separate from the syntax, the lexical point of view examines the vocabulary of the language. Lastly there is the semantic point of view, which examines the meaning of the entire requirement statement [7].



Similar to the model presented by [7], NASA's SATC has presented a quality model to grade engineering requirements [10]. SATC has a compiled list of desirable characteristics for engineering requirements which are listed in Table 2.5.

**Table 2.5 Quality attributes from SATC [10]**

<b>Complete</b>	A complete requirements specification must precisely define all the real world situations that will be encountered and the capability's response to them
<b>Consistent</b>	A consistent specification is one where there is no conflict between individual requirement statements that define the behavior of essential capabilities; and specified behavioral properties and constraints do not have an adverse impact on that behavior.
<b>Correct</b>	For a requirement specification to be correct it must accurately and precisely identify the individual conditions and limitations of all situations that the desired capability will encounter and it must also define the capability's proper response to those situations
<b>Modifiable</b>	In order for requirements specifications to be modifiable, related concerns must be grouped together and unrelated concerns must be separated
<b>Ranked</b>	Ranking specification statements according to stability and/or importance is established in the requirements documents' organization and structure.
<b>Testable</b>	In order for a specification to be testable it must be stated in such a manner that pass/fail or quantitative assessment criteria can be derived from the specification itself and/or referenced information
<b>Traceable</b>	Each statement of a requirement must be uniquely identified to achieve traceability. Uniqueness is facilitated by the use of a consistent and logical scheme for assigning identification to each specification statement within the requirements document.
<b>Unambiguous</b>	A statement that specifies a requirement is unambiguous if it can only be interpreted one way.
<b>Understandable</b>	A requirements specification is understandable if the meaning of each of its statements is easily grasped by all of its readers.
<b>Validatable</b>	In order to validate a requirements specification each of the individuals and organizations having a vested interest in the system solution must be substantiate that the requirements are true as stated
<b>Verifiable</b>	In order to be verifiable, requirement specifications at one level of abstraction must be consistent with those at another level of abstraction

The quality attributes presented above, relate to both individual requirements and the entire requirement documents. In order to analyze individual and the entire requirement document SATC developed a list of nine objective quality indicators that represent the above quality attributes.

**Table 2.6 Quality indicators from SATC [10]**

<b>Individual Requirements</b>	Imperatives
	Continuances
	Directives
	Options
	Weak Phrases
<b>Entire Document</b>	Size
	Specification Depth
	Readability
	Text Structure

To keep within scope of the presented research, only the quality indicators for individual requirements will be further examined. Imperatives are words and phrases that show a level of necessity [10], directives point to illustrative elements in the requirement document [10], options give the designer flexibility in satisfying the requirements [10], weak phrases are phrases or words that tend to cause uncertainty and ambiguity in a requirement statement [10]. The weak phrases will be an area of focus that this research will attempt to address [10].

**Table 2.7 Examples for the imperatives used by SATC [10]**

<b>Imperatives</b>	shall
	must
	is required to
	are applicable
	responsible for
	will
	should
<b>Continuances</b>	below
	as follows
	following
	listed
	in particular
	support
<b>Directives</b>	figure
	table
	for example
	note
<b>Options</b>	can
	may
	optionally
<b>Weak Phrases</b>	adequate
	as a minimum
	as applicable
	easy
	as appropriate
	be able to
	be capable
	but not limited to
	capability of
	capability to
	effective
	if practical
	normal
	provide for
	timely

### 2.2.3 Requirement Modeling

Representations other than natural language that are used to represent engineering requirements include (UML) and SysML which are formalized representations of NL requirements. Formal representations limit the structure and provide a systematic way of expressing a requirement. This, in turn, helps to limit ambiguous, inconsistent, and incorrect requirements. However, this benefit does come at the expense of usability and expressiveness. Formal representations, such as UML and SysML, enable requirements sentences to be modeled and relationships between individual requirements to be captured. UML is a standardized object oriented language comprised of several models used in abstraction of the system [18]. UML consists of three models a state model to represent the behavior, an interaction model to represent the collaboration of individual objects, and a class model to represent the structural data aspects of a system [18]. SysML is an extension of UML that uses many of the same diagrams as UML 2.0. The additional diagrams added were the Requirements diagram and the parametric diagrams [19]. Both UML and SysML require some form of training to be used properly, which is why requirements are still mainly expressed using natural language.

### 2.2.4 Current Analysis Tools

To counter ambiguity, inconsistency, and incorrectness, natural language processors (NLP) are used to parse natural language requirements to transform any ambiguous or incorrect requirements into correct requirements. QuARS (quality analyzer for requirements specifications) is a tool being developed that analyzes requirements for linguistic correctness. QuARS uses linguistic analysis to examine the lexical and

syntactical properties of an engineering requirement. It uses a syntax parser in combination with a lexical parser to identify defects in requirements that are related to the quality indicators, vagueness, subjectivity, optionality, implicity, weakness, under-specification, multiplicity, and readability [7]. The defects are identified by using property related dictionaries that contain the words and syntactical elements that identify with the quality indicators [7].

QuARS and other NL processing tools aids in determining the expressiveness of a requirement statement by parsing requirements for particular keywords that relate to the previously mentioned quality measures. This is a pragmatic approach that is effective at identifying quality measures only within controlled natural language requirements. Controlled natural languages are a subset of the unrestricted natural language that places limitations on the available vocabulary, the syntax, and the semantics. By using a controlled NL, analysis issues pertaining to the following can be avoided [12]:

- System may not be able to parse a sentence because the system may only consist of a finite number of lexical entries.
- The system may apply more than one POS tag to words within the requirement.
- Analyzing the semantic meaning of an unrestricted language is difficult to achieve programmatically.

The use controlled natural language to express requirements is not without its limitations [12].

- Controlled natural languages can become too restrictive becoming more like a formal representation

- Guidance for the user to create meaningful requirements is needed

A way to deal with the problems created by the use of a controlled natural language is to limit the vocabulary syntax and semantics as minimal as possible. A controlled natural language is avoided as an option to assist the grammatical and linguistic analysis of engineering requirements because it limits the design domain of the designer.

### **2.3 Research Questions**

From the literature review it was determined there exists a need for a formalized syntax for engineering requirements based on linguistics instead of pragmatic examples. The objective quality indicators presented by [7] and [10] are based on a finite list of terms that have been identified to relate to the quality attributes. Words or phrases that are not in the finite list that still may relate the quality attributes could lead to falsely identified correct requirements. A rule based system derived from linguistics and grammar can possibly avoid the downfalls of finite lists of terms.

Also notably absent from the literature is a method to linguistically classify types of engineering requirements. Classifying engineering requirements will assist in the analysis because it narrows the focus of how to analyze engineering requirements. All engineering requirements do not consist of the same elements; therefore they need to be analyzed differently. These missing elements in the field of requirements engineering lead to the development of the research questions below:

1. Based on grammar and linguistics, what is the structure for expressing engineering requirements?
2. How can engineering requirements be classified using linguistics and grammar?
3. What types of analysis on engineering requirements can be completed using the formalized structure and taxonomy rules developed from linguistics and grammar?

To answer the research questions stated above, the thesis will be organized as following: Chapter three will examine the linguistic and grammatical elements of a NL requirement statement. This serves as the foundation of the research. Chapter four discusses the formalized syntax for expressing an engineering requirements based on linguistics and grammar. Chapter five consists of examining the proposed syntax and analysis method with the use of a test case. To examine the proposed syntax the next section applies the proposed syntax and rules to a requirements document. The thesis will conclude with a section that will detail what should be done to advance the research on using linguistics and grammar to express and analyze engineering requirements.



## CHAPTER THREE: FUNDAMENTAL BASIS FOR EXPRESSING AND ANALYZING ENGINEERING REQUIREMENTS

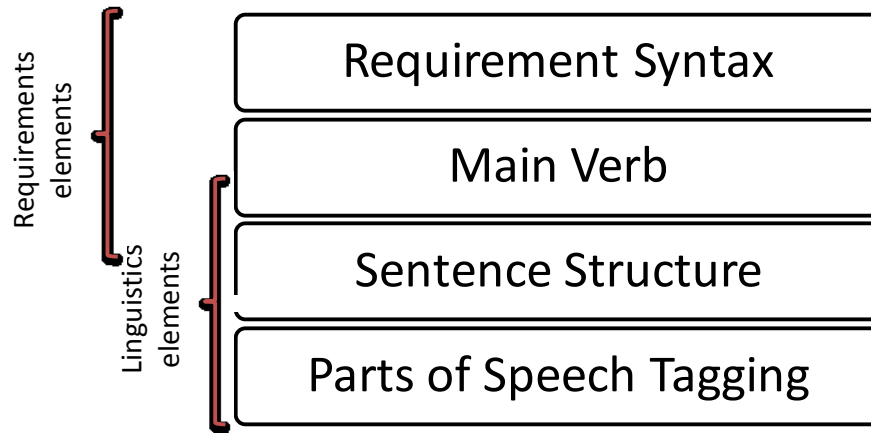
This section describes the underlying linguistic elements used to express and analyze engineering requirements. The method proposed in this section aims to provide a more fundamental approach to objectively identify and express functional, non-functional, qualitative, and quantitative requirements. The method is based on NL grammatical and linguistic rules that lead to well expressed requirement statements. By examining the linguistic and grammatical rules that lead to well expressed requirements, a method that does not limit the vocabulary available to designers can be developed to express engineering requirements.

### 3.1.1 Linguistic Approach

The method proposed uses a linguistic approach that analyzes the parts of speech, sentence structure, and verb types to determine the elements of a well expressed engineering requirement statement. By identifying the linguistic and grammatical elements that make up a well written requirement statement, it is hypothesized that ambiguity and incompleteness, within the statement will be reduced. In addition to improving the quality attributes discussed in the previous chapter, the linguistic analysis of requirement statements should result in objectively being able to classify the requirement type (i.e. functional and non-functional).

A three tiered linguistic approach will be used to analyze the requirement statements. The figure below represents these three tiers, which identifies the parts of speech, sentence structure, and main verb type used in a requirement statement. This

linguistic analysis then results in the classification of the requirement statement and the identification of possible errors within the requirement statement.



**Figure 3.1 Four tiered approach to linguistic analysis of engineering requirements**

The first tier is based on Parts of Speech tagging, which identifies the different types of words that exist within a natural language requirement statement. The second tier is based on sentence structure, which shows how the tagged words relate to one another. This tier takes into account the context of the words used in the requirement statement. The third tier examines the grammatical verb type in the sentence, which is based on the three previous tiers. This step begins the classification of engineering requirement statements. Finally, these grammatical tiers lead up to a syntax that is specialized for functional and non-functional engineering requirement statements.

### 3.1.2 Parts of Speech (POS) Analysis

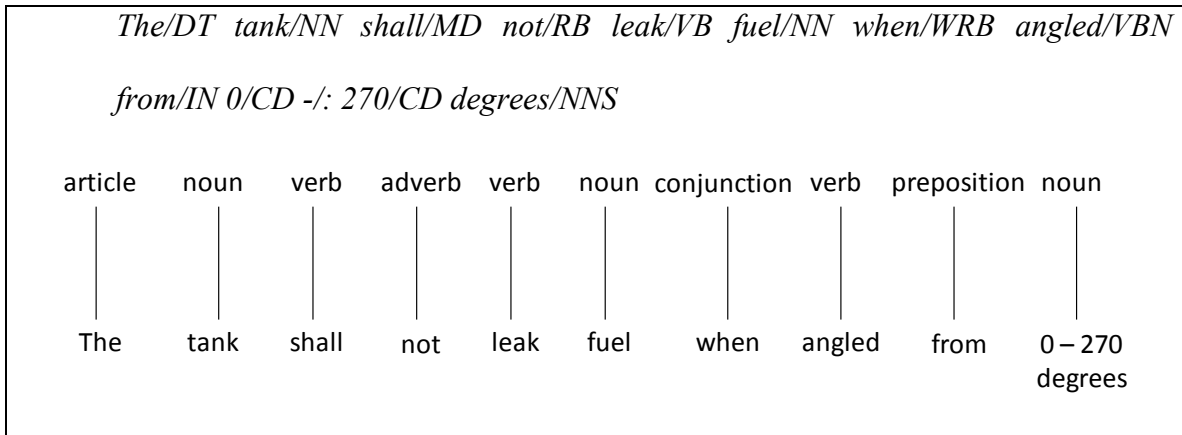
The first step in the process of analyzing requirements from a linguistic viewpoint is identifying the parts of speech being used in well expressed requirement statements. Traditional grammar has eight parts of speech: nouns, pronouns, verbs, adjectives, adverbs, prepositions, conjunctions, and interjections [20]. These are formally defined in the table below.

**Table 3.1 Eight parts of speech of traditional grammar [20]**

<b>Parts of Speech</b>	<b>Definitions</b>
<b>Nouns</b>	Words that name something either intangible or tangible.
<b>Pronouns</b>	Words used as a substitute for a noun or, sometimes, another pronoun.
<b>Verbs</b>	Shows the performance or occurrence of an action or the existence of a condition or a state of being.
<b>Adjective</b>	Modifies a noun by describing it more definitely or fully or by narrowing a nouns meaning.
<b>Adverb</b>	An adverb is a word that qualifies limits, describes, or modifies a verb, an adjective, or another adverb.
<b>Preposition</b>	A word or phrase that links an object (a noun or noun equivalent) to another word in the sentence to show the relationship between them.
<b>Conjunction</b>	Connects sentences, clauses, or words within a clause.
<b>Interjection</b>	Also known as an exclamation is a word, phrase, or clause that denotes strong feeling

An example requirement that has been tagged is presented in Figure 3.2. The first part of the figure has been tagged with standard tags from Penn Treebank tag set. This set of tags is based on the Brown University Standard Corpus of Present-Day American English (Brown Corpus) [21]. The first section of the figure has been tagged with the Penn

Treebank tag set and in the second section of the figure, the abbreviations in the Penn Treebank tag set has been translated to the actual words to make the tags more apparent.



**Figure 3.2 A tagged requirement statement using the Penn Treebank tag set**

This type of analysis can be achieved programmatically using parts of speech taggers such as the one developed by Stanford University’s Natural Language Processing Group. Stanford’s tagger is a corpus based log linear part of speech tagger [22]. A log linear conditional probability model derived from previously tagged text is used to train the tagger. This helps to improve the accuracy of the tagger by estimating the probability of a tag sequence [22].

As previously noted, POS tagging applications exist, but are not always correct. Shown in Table 3.1, are inconsistencies within tagged sentences. These errors stem from the POS tagger not being able to capture the relationships between words and the context of the words. However, tagging a sentence is still useful as it allows for some analysis of engineering requirement statements. Thus in this research, requirements were tagged

manually and iteratively based on sentence structure and grammatical functions, presented in Section 3.1.2 and 3.1.3.

**Table 3.2 Examples of inconsistencies within tagged sentences**

Carl must prune the peach tree. Carl/NNP must/MD prune/VB the/DT peach/NN tree./NN
Carl must prune the red tree. Carl/NNP must/MD prune/VB the/DT red/JJ tree./NN
Carl must prune the orange tree. Carl/NNP must/MD prune/VB the/DT orange/NN tree./VBZ
The vehicle is orange. The/DT vehicle/NN is/VBZ orange./JJ
The vehicle is red. The/DT vehicle/NN is/VBZ red./JJ

### 3.1.3 Sentence Structure

To further analyze engineering requirement statements, sentence structure is examined to add word context as an element to the NL requirement analysis process. The previously mentioned parts of speech are the underlying foundation in sentences, but they only offer a limited amount of information about a requirement. By examining the sentence structure, the requirement statement can be separated into grammatical functions which show how the parts of speech are related to one another syntactically. There are

four main types of grammatical functions: subjects, objects, complements, and adjuncts [23], which are detailed in Table 3.3. By definition the subject, object, and complements are the essential elements of a sentence with adjuncts being optional [23].

**Table 3.3 Grammatical functions [23]**

<b>Subject</b>	<b>Subjects</b>		One of the two main parts in a sentence (other being the predicate). A noun phrase that tells what the sentence is about
	<b>Verb</b>		
<b>Predicate</b>	<b>Objects</b>	<b>Direct object</b>	The thing acted on by a verb
		<b>Indirect object</b>	The first noun phrase when a verb is followed by two noun phrases
	<b>Complements</b>	<b>Subject Complement</b>	Necessary element of a sentence that completes the subject of a sentence that uses a linking verb
		<b>Object Complement</b>	Necessary element of a sentence that provides more information about the object
		<b>Adverbial Complement</b>	Adverbial phrases that is a necessary element of a sentence
	<b>Adjuncts</b>		Word or phrase that is used to modify another word or phrase. Adjuncts are typically optional

The two highest level parts of a sentence are the subject and predicate [23]. The subject is identified as a type of grammatical function itself and is identified as a noun phrase. The subject can be classified in three ways: grammatical, logical, and thematic [23]. Grammatical subjects (G) are required in all sentences and always precede the main

verb. Logical subjects (L) are determined by taking into consideration the syntax and semantics of the sentence. Logical subjects usually describe the participant in a sentence. Thematic subjects (T) are what the sentence is about [23]. For simplicity and to avoid ambiguity within engineering requirements, the subject of a requirement should be all three subject types. This reduces the sentence to only one subject that will always be at the beginning of the requirement statement. Below are example sentences from [23], which demonstrate the three subject types.

*Ex 3.1* John (G, L, T) took the largest kitten

*Ex 3.2* The largest kitten (G, T) was taken by John (L)

*Ex 3.3* The largest kitten (T), we (G, L) gave away

The predicate is the second main element of a sentence and expresses what is said about the subject. Since the predicate itself is not a grammatical function it must be broken down into its grammatical function elements. The predicate always consists of at least a verb, but is often accompanied by objects, subject complements, object complements, adverbial complements, or adjuncts [23].

An object is the thing acted on by the main verb in a sentence. Similar to the subject, the object consists of a noun phrase. The object always follows the main verb, it is not in construction with a preposition, and it is a necessary element with transitive verbs. Below is an example of an object in an engineering requirement.

*The user operates the Vehicle(D.O.)*

Complements are elements of a sentence that assigns attributes to the subject, helps to further identify the subject, or places a location to the subject. These types of complements are referred to as attributive, locative, or identity complements, respectively. Three examples are given here to illustrate these distinctions.

*The car is red*

*The car is in the red zone*

*The car is the red one*

In the first example above, an attributive complement is used. It is assigning the color red as a characteristic of the car. In the second example the complement is in the form of a prepositional phrase and it is used to place a location to the subject. The third example uses an identity complement which is used to distinguish the car. Attributive complements correlate directly to the definition of a requirement as being a statement that describes characteristics of a system.

Adjuncts are typically adverbial phrases and are an optional element of a sentence. Adjuncts tend to describe things such as time, place, extent, and manner and add and are not bound by location within a sentence [23]. In engineering requirements, adjuncts add detail to the requirements, affecting the complexity of an engineering requirement statement. The added level of detail adjuncts add to a requirement is shown in the examples below.



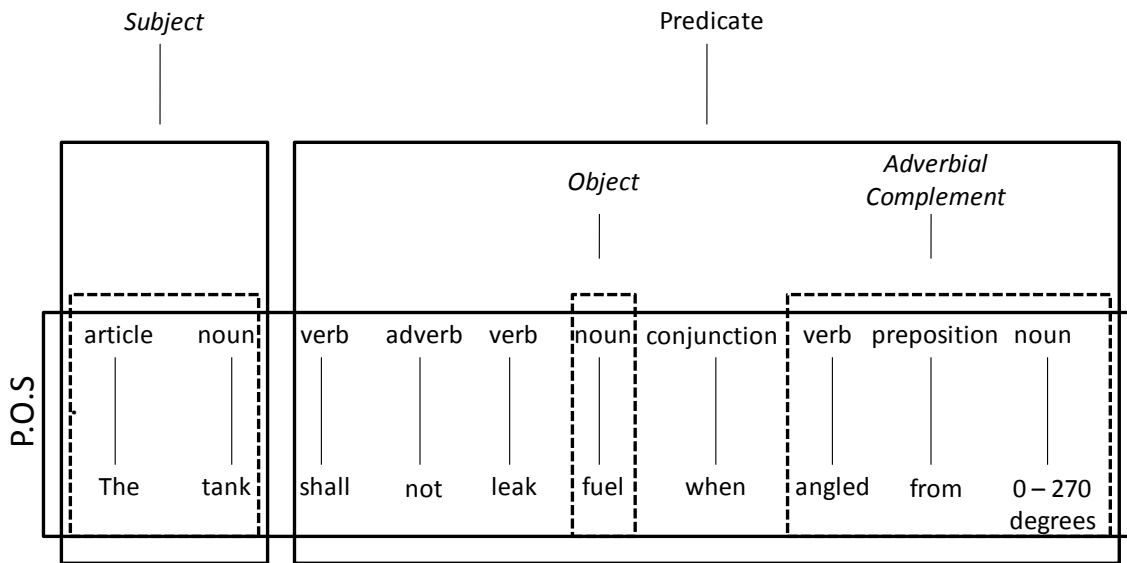
*The car must accelerate*

*The car must accelerate quickly*

*The car must accelerate from 0-60 in 4.5 seconds*

The first requirement does not contain an adjunct, leaving the parameter of acceleration unknown. The second requirement uses the adverb quickly to describe the acceleration. This adds detail to the requirement but quickly can be interpreted multiple ways depending on the reader. The third requirement uses the prepositional phrase “from 0-60 in 4.5 seconds” to add enough to detail to ensure the requirement is disambigous. The appropriateness of the requirements in the example is determined by the stage at which the requirement statement was written. The first two requirements may be expressed as high level requirements that occur early in the design process and are transformed into more specific requirements as in the third requirement.

Figure 3.3 shows the grammatical functions being applied to a single engineering requirement.



**Figure 3.3 Subject and predicate**

Other than the subject, all the other components of a sentence are included in the predicate. As stated earlier the most basic element in the predicate is the main verb. Identifying the main verb in a requirement sentence is a key component in analyzing engineering requirements which is needed to classify the requirement as functional, non functional or a user requirement. There are three main verb types used in English grammar: action, linking, and helping verbs.

Action verbs describe an action or behavior and are either transitive or intransitive. Transitive verbs are action verbs in which an object receives the action. The object receiving the action is always a noun.

<i>Ben</i>	<i>rides</i>	<i>his</i>	<i>bike</i>
<subject>	<verb>	<direct object>	

In the example above *Ben* is the subject *ride* is the verb and *bike* is the noun that is receiving the action from the verb. The combination of the verb type and direct object makes this requirement statement transitive.

Often lumped into the category of action verbs are verbs that show possession. Verbs such as *have*, *include* or *possess* are possessive verbs that influence the nature of a requirement statement differently. These types of verbs require that something must exist within the system being designed. Therefore, when these types of verbs are used in a requirement statement they often describe how the system must be designed instead of what the system must do. In the example below the *chair* is the system being designed, the main verb is *have* and the *cushion* represents how the system will be designed. This requirement does not state the purpose of the cushion.

<i>The</i>	<i>Chair</i>	<i>must</i>	<i>have</i>	<i>a</i>	<i>cushion</i>
	<subject>	<modal>	<possessive verb>		<direct object>

The second form of an action verb is intransitive. An intransitive verb is one that never has an object receiving the action of the main verb. In the example below the *light* is the subject and *shine* is the intransitive verb. The second sentence contains the adverb, *brightly*, that modifies the action verb, *shines*. In other words, the adverb provides specific information pertaining to how the light performs.

<i>The</i>	<i>light</i>	<i>shines</i>
	<subject>	<verb>

<i>The</i>	<i>light</i>	<i>shines</i>	<i>brightly</i>
	<subject>	<verb>	<adjective>

Linking verbs do not show action, instead they connect the subject of a sentence to a noun or adjective that describes a state or property of the subject. The noun or adjective that is being connected to the subject is referred to as the subject complement.

<i>The</i>	<i>light</i>	<i>is</i>	<i>hot</i>
	<subject>	<verb>	<adjective>

In the above examples the subject of the sentence is *light* and the linking verb is *is*. In the above example, the adjective “hot” is the subject complement.

Helping verbs are used in conjunction with action or linking verbs to add additional information or apply constraints on the subject. Helping verbs are categorized into two categories: primary and modal. The primary helping verbs are forms of be, do, and have, which can also be used as main verbs.

<i>The</i>	<i>light</i>	<i>is</i>	<i>illuminating</i>	<i>the</i>	<i>room</i>
	<subject>	<helping	<verb>		<direct

verb>

object>

Modal verbs, which are more relevant to engineering requirements, modify the semantics of the main verb by showing level of necessity or possibility. The ten modal verbs are shown below in Table 3.4.

**Table 3.4 Common Modal Verbs**

Can	May	Must	Shall	Will
Could	Might	Ought to	Should	would

In the example below, must is used as the modal verb in the sentence. The levels of necessity of the ten modal verbs are often subjective and determined by the reader.

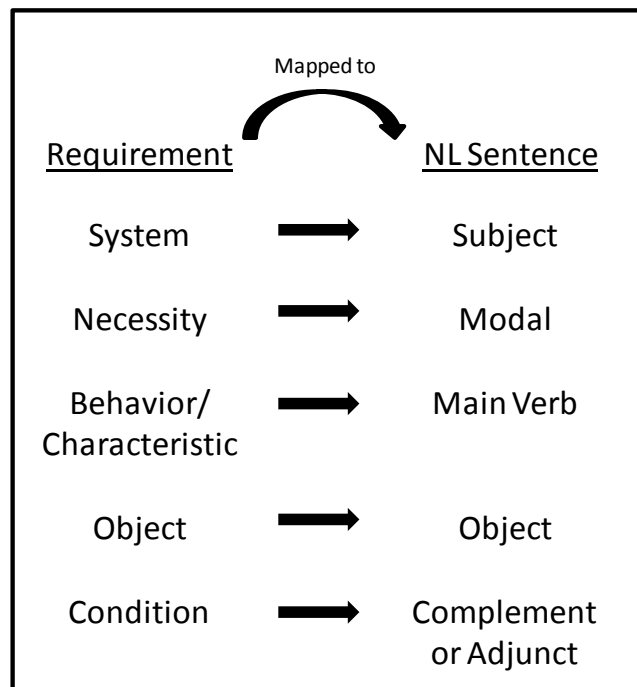
*The light must illuminate the room*

<subject>      <modal>      <verb>      <direct object>

Determining the verb types used in engineering requirements is essential in analyzing engineering requirements. The requirement will be analyzed differently depending on the type of requirement which will be shown in the next chapter. The next section builds upon the linguistic elements mentioned in this section and aims to create syntax for expressing engineering requirement statements and an analysis method to process requirement statements.

### **3.2 Grammar and linguistics directly applied to requirement statements**

As defined earlier, an engineering requirement is a statement that describes a condition or capability that must be met or possessed by a system or system component. This implies that a requirement statement must include a system, a word or phrase that demonstrates the level of necessity that condition or capability must be met, a phrase that describes that condition or capability. These attributes of an engineering requirement can be described with four key components: system, necessity, behavior or characteristic, and a condition. The NL elements described in the previous section are shown being mapped to their respective requirement elements in Figure 3.4. The syntax for engineering requirements will be based on this mapping.



**Figure 3.4 Mapping of NL requirement elements to NL sentence elements**

### 3.2.1 System

In an engineering requirement the system being designed should be the subject of a NL sentence. This will remove a level of ambiguity within a requirement statement. As stated in the previous section, there are different types of subjects within a NL sentence: grammatical, logical, and thematic. In order to remove ambiguity about the system being designed, the subject of an engineering requirement will be all three subject types. This requires that the requirement be in the same format as example 3.1, where the subject will always precede the other components in the requirement, it will be what the requirement is about, and it will describe the main participant in the requirement. In Figure 3.5, the gasket is what the sentence is about, it is the main participant in the sentence, and it precedes all the other components of the sentence. When a requirement is expressed in this manner, the artifact being designed is clearly expressed for the reader, removing a level of ambiguity from the requirement.

<u>Subject</u>	<u>Predicate</u>
John (G, L, T)	took the largest kitten
The gasket (G, L, T)	must prevent water contamination

**Figure 3.5 Subject organization within a requirement [23]**

Although the subject is a noun, parts of speech taggers and other NL processing have difficulties identifying the subject of a sentence. This is due to the fact that parts of speech taggers do not take into account the context of the words in a sentence. Currently,

the only method to ensure that a requirement is structured in the manner stated above is with human reasoning. To programmatically achieve this, a rule based system needs to be developed as another element to the NL processing tools.

### 3.2.2 Necessity

The necessity of a requirement describes the level of importance of the requirement and in a sense makes a requirement indeed a requirement. Authors such as Pahl and Beitz refer to this in binary as either a demand or wish [3]. The necessity of a requirement is expressed using modal auxiliary verbs such as shall and must. These modal verbs directly follow the subject as shown in the example below.

*The gasket <must> prevent water contamination*

Depending on the requirements document, the auxiliary modal verbs may have different levels of importance associated with them. Currently there is no agreed upon level description of the modals shown in Table 3.4. If levels of importance can be associated with the modal verbs, this would improve the consistency of requirements by allowing the reader to know how important the requirement is to the system being designed.

The necessity element of a requirement can be determined by NL processing tools since it is a modal which is a part of speech itself. By identifying and keeping track of the modals being used throughout a requirements document, a NL processing tool should be able to ensure the consistent use of the modals being used to show levels of necessity.



### 3.2.3 Behavior / Characteristic

The function of a requirement describes the action or behavior of the system. The state of a requirement describes the characteristics of the system. From examination of many requirements and based on the syntax of NL, the function or state is always located after the modal and is represented by a verb [23]. This verb is the main verb of the requirement and can be either an action verb describing functionality or a linking verb that links properties to the subject. As stated in the previous section there is three main verb types: transitive, intransitive, and linking. As action verbs are either transitive or intransitive verbs, these verb types are used to describe functionality and behavior. Linking verbs are used to describe a state by linking properties to the system. Examples of these verb types in a requirements statement is shown below.

<i>The</i>	<i>seat</i>	<i>must</i>	<i>prevent</i>	<i>injury</i>		
	<subject>	<modal>	<trans. verb>	<direct object>		
<i>The</i>	<i>airplane</i>	<i>seat</i>	<i>must</i>	<i>float</i>		
	<subject>		<modal>	<intrans. verb>		
<i>The</i>	<i>seat</i>	<i>must</i>	<i>be</i>	<i>Easy</i>	<i>to</i>	<i>adjust</i>
	<subject>	<modal>	<linking verb>	<subject complement>		

Since NL processing tools are based on Brown Corpus' set of tags, they can only identify verbs and their tenses, and not their type (i.e. transitive, intransitive, linking). Identifying the verb type is a key step in analyzing natural language requirements as they assist in objectively classifying the requirement as functional or non functional.

### 3.2.4 Condition

The condition of a requirement statement answers the question, how much or to what extent. In linguistic terms the condition in a requirement statement is either a subject complement or an adjunct. Subject complements are necessary for requirements where the main verb is linking. In the example below “water resistant” is the subject complement as it is necessary information about the subject. Without the subject complement the requirement will be incomplete. The purpose of the subject complement is to assign characteristics to the system or system component and the characteristic that is being assigned is the gaskets being water resistant.

<i>The</i>	<i>gaskets</i>	<i>must</i>	<i>be</i>	<i>water</i>	<i>resistant</i>
	<subject>	<modal>	<linking verb>	<subject complement>	

To answer the question, to what extent, an adjunct is appended to the sentence. This information is not necessary and the requirement is complete without it. However, as requirements are refined during the design process, adjuncts gain more importance as they help to distinguish low level requirements from high level ones. An example of an adjunct being used in a non functional requirement is shown below.



- Based on verb type classify requirement as functional or non-functional

To enable consistency among requirement statements a formalized syntax and analysis method will be proposed based on the linguistic elements that aims to improve the aforementioned quality properties. The syntax and analysis method will be discussed in the next chapter.

## CHAPTER FOUR: REQUIREMENT SYNTAX AND ANALYSIS METHOD

The linguistic approach presented in this chapter, is based on the grammatical and linguistic elements discussed in the previous chapter. A NL requirement syntax was developed by organizing the grammatical functions and parts of speech, which is shown in Table 4.1, into a formalized structure that can be used to represent and classify engineering requirement statements. This is different from the boilerplates discussed by Hull [6], as this approach is based on linguistics and enables all engineering requirements to be classified. Two different specific types of requirements are classified, namely functional and non-functional requirements.

**Table 4.1 Grammatical and linguistic elements that are used as the foundation of the NL requirement syntax**

<b>Grammatical Functions</b>	<b>Parts of Speech</b>
Subject	Noun
Direct Object	Noun
Subject Complement	Adjective or nouns
Adjunct	Prepositional phrases
-	Verbs (intransitive, transitive, linking, and modal)

### **4.1 Syntax**

Within the scope of this thesis, two main types of requirements have been identified, functional, non-functional. In order for the linguistic and grammatical elements to be consistently applied to requirement statements, a formalized syntax based on the elements is needed. By observing the individual constituents of a requirement

statement rather than observing the requirement statement as a single entity such as in UML and SysML, requirements can be analyzed more effectively.

Backus-Naur Form (BNF), a syntactic meta-language often used to express the domain of formal language, is used to express the syntax. The benefits of using a meta-language to describe the syntax are the following [24]:

- It defines the various parts of a syntax
- It shows all the syntactically valid sequences of symbols of a syntax
- It shows the syntactic structure of any sentence of the language

BNF defines the syntax by using a set of rules that defines all of the possible forms of the syntax starting with the terminal terms and uses these to describe the non-terminal terms. Terminal terms are denoted by double quotation marks, and non terminal terms are denoted with brackets (< >). Non-terminal terms show up as symbols that represent the purposeful organization of terms within the syntax. The symbol is always to the left and the syntactic terms or other symbols to describe it are on the right. Two sets of colon and an equal sign are used to separate the symbol from the descriptive syntactic terms. A vertical bar is as an operator that denotes a choice can be made.

**Table 4.2 BNF general notation**

<b>BNF symbol</b>	<b>Description</b>
“ “	Terminal term
< >	Non-terminal term or symbol
	Shows that a choice can be made
{ }	Optional element in the syntax
::=	Separates the symbol from the expression

For the syntax developed for NL requirement statements, the non-terminal terms are the parts of speech and the highest level term is a general requirement. At the most fundamental level a NL requirement has a system represented by the subject, a modal that shows necessity, a verb phrase that either shows action or provides a means to link characteristics to the system, and also includes any condition that the system must meet. The subject is the focus of what is being designed and is represented by a noun phrase. The system's behavior and characteristics are represented by the verb phrase. The verb phrase also has direct objects, subject complement, and adjuncts that help to describe the extent of the behavior or characteristics. The condition explains the extent of the system's behavior or characteristics.

**Table 4.3 BNF of general requirement**

<b>Requirement Terms (General)</b>	<b>Linguistic Terms</b>
<requirement> ::= “system” “necessity” “behavior/characteristics” “condition”	<requirement> ::= <subject> “modal” <verb phrase>
System	<subject> ::= <noun phrase>
Behavior/Characteristic	<verb phrase> ::= “intransitive verb”   “transitive verb”   “linking verb” {<direct object> <subject complement> <adjunct>} <object> ::= <noun phrase>
	<complement> ::= <noun phrase>   “adjective”
Condition	<adjunct> ::= <prepositional phrase>
	<noun phrase> ::= “article” “noun”

To further detail NL requirement statements, the syntax will depict the structure of functional requirements and non-functional requirements. Functional requirements represent behaviors of a system or system component and can be represented grammatically two different ways. The first representation uses an intransitive verb as the main verb type to represent the behavioral aspect of the requirement. The second representation for functional requirements uses transitive verbs to as the main verb type to represent the behavioral aspect. When the main verb is transitive a direct object must follow, which can be considered an element in the condition of the requirement. To add additional conditions to the requirement statement, adjuncts are appended to the statement.



**Table 4.4 BNF of functional requirement**

<b>Requirement Terms (Functional)</b>	<b>Linguistic Terms</b>
<functional requirement> ::= “system” “necessity” “behavior” “condition”	<functional requirement> ::= <subject> “modal” <main verb> {<direct object>} {<adjunct>}
System	<subject> ::= <noun phrase>
Behavior	<main verb> ::= “intransitive verb”   “transitive verb”
Condition	<direct object> ::= <noun phrase>
Additional condition(s)	<adjunct> ::= “prepositional phrase”
	<noun phrase> ::= “article” “noun”

Non-functional requirements represent purely characteristics of a system or system component. Non-functional requirements are represented using linking verbs as the main verb. Linking verbs are followed by a subject complement which represents the characteristic the non-functional requirement is representing. As with functional requirements, a level of detail can be added to the requirement with the addition of an adjunct which represents the condition.

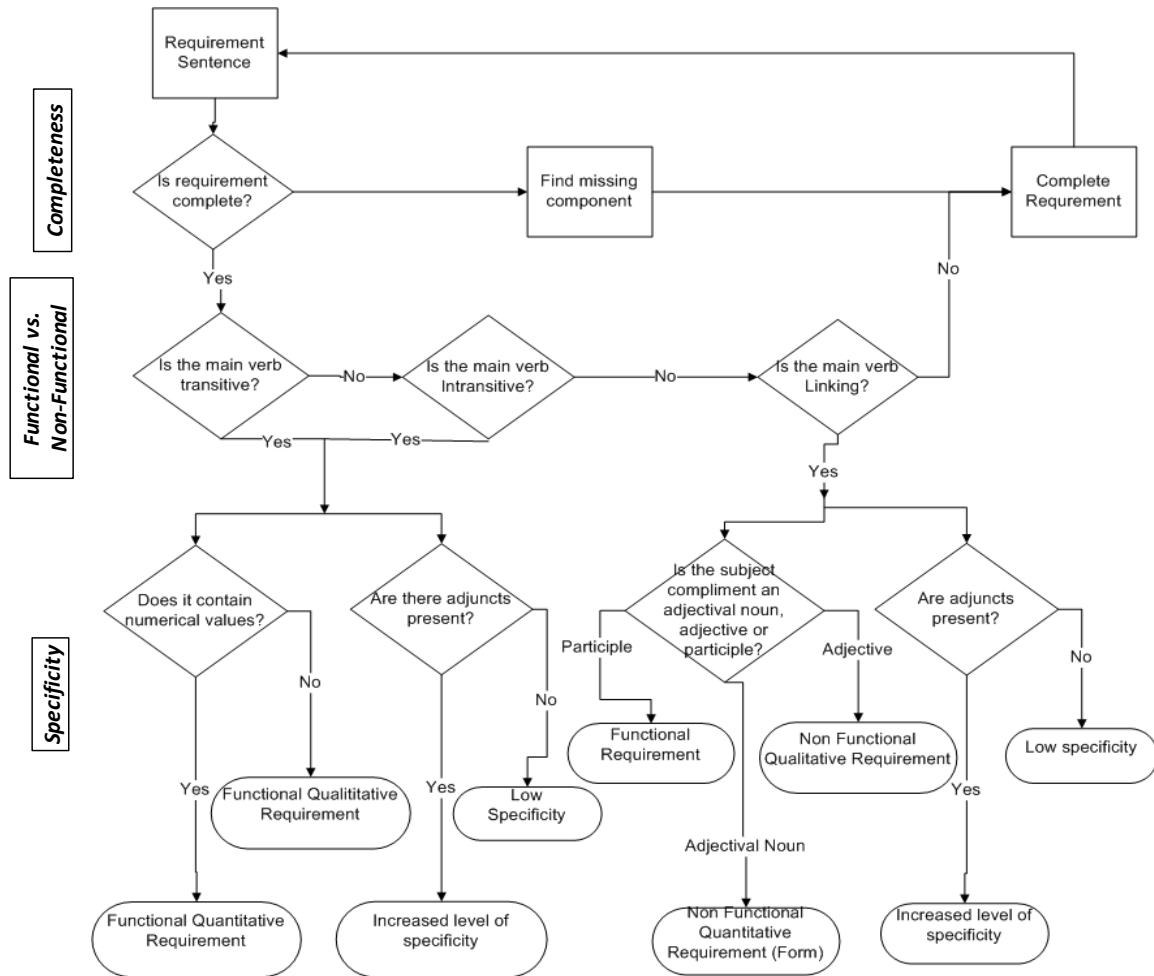
**Table 4.5 BNF of non-functional requirement**

<b>Requirement Terms (Non-functional)</b>	<b>Linguistic Terms</b>
<non-functional requirement> ::= “system” “necessity” “characteristic” {“condition”}	<non-functional requirement> ::= <subject> “modal” “linking verb” <subject complement> {<adjunct>}
System	<subject> ::= <noun phrase>
Characteristic	<subject complement> ::= <noun phrase>   “adjective”
Condition	<adjunct> ::= “prepositional phrase”

## **4.2 Linguistic Analysis Process**

The process used to analyze engineering requirement statements linguistically is based on the syntax described in the previous section. In addition to determining whether

a requirement is functional or non-functional, it is hypothesized that the analysis process based on the syntax can objectively determine the completeness, specificity, qualitativness, and quantitavness of engineering requirements.



**Figure 4.1 Flow chart describing linguistic analysis process**

From Figure 4.1, the first step in analysis process is checking an individual requirement statement for completeness. Complete requirement statements contain all of the necessary linguistic elements of that specific requirement type (Table 4.3). In the

examples shown below, the first statement is missing the subject and modal, therefore the system being designed is not known nor is the level of necessity. In the second statement the subject and modal has been added to make a complete requirement.

*Allow for forward travel - Incomplete*

*Seat must allow for forward travel – Complete*

The second step is to classify the main verb of the requirement statement as transitive, intransitive, or linking. Identifying the verb type determines whether the requirement is functional or non-functional. If the main verb is transitive or intransitive, the requirement has a good chance of being functional. If the main verb is linking the requirement is likely to be non-functional.

*The crank must <rotate> at 3000 rpm - Intransitive*

*The crank must <transmit> torque - Transitive*

*The crank must <be> durable - Linking*

From the flow chart, adjuncts are the next linguistic element to be identified. Since adjuncts add additional information to the requirement, their presence within a requirement statement is indicative of the specificity of the requirement. The number of adjuncts in a requirement is directly proportional to the requirement's level of specificity. This is demonstrated in the examples below. The first requirement has no adjunct and is

the most abstract requirement. The second requirement adds one adjunct that adds how fast the crank must rotate. In the third requirement, the speed at which the crank must rotate is known, and it also adds specifics on the noise level of the rotating crank.

*The crank must rotate*

*The crank must rotate <quickly>*

*The crank must rotate <quickly> <with a minimum noise level>*

In addition to adjuncts, the existence of numerical values also adds to the level of specificity within a requirement statement. The count of numerical values and count of adjuncts are not mutually exclusive when determining how specific the requirement is. The specificity of a requirement increases as the count of numerical values in requirement statement reaches the count of adjuncts in the statement.

*The crank must rotate*

*The crank must rotate at <3000> rpm*

*The crank must rotate at <3000> rpm with a minimum noise level of <90> db*

If a functional requirement does not contain any numerical values the requirement has a high probability of being qualitative.

Requirement statements where the main verb is linking are analyzed differently because of the different linguistic elements that make up such requirements. After the

main verb has been identified as being linking, the type of subject complement used in the requirement is identified. The subject complement is identified as being an adjectival noun, adjective or participle. An adjectival noun is a noun that operates as an adjective. An example is shown below:

*The tube must be <red>*

In the example, the noun red which is considered a noun is operating as an adjective to modify the tube.

A participle is the adjectival form of a verb that assigns the characteristic of the verb to the subject. An example is shown below:

*The user should be <protected>.*

The presence of adjectival nouns has a high probability of producing form specific non functional requirements, while the presence of adjectives tends to result in qualitative non-functional requirements. Linking verb requirements where the subject complement is a participle act similar to intransitive functional requirements.

### **4.3 Quality Attributes Addressed by Analysis Method**

The syntax introduced in the previous section addresses two main quality attributes of engineering requirements, completeness, and specificity.

#### 4.3.1 Completeness

A complete requirement should fully express distinct actions, behaviors, or characteristics of an artifact being designed. The proposed syntax achieves this by

making the designer aware of the essential elements of a requirement sentence. The syntax presented addresses the elements for, functional and non-functional requirements.

Functional requirements describe a function or behavior of the artifact being designed, making them either transitive or intransitive NL sentences. Transitive requirements must consist of an artifact, necessity verb, an action verb, and a direct object for them to be complete requirements. These four components allow the reader to know exactly what is being designed, the level of importance, the functionality or behavior of the artifact, and what the artifact is affecting. To add additional information about the artifact being designed, adjuncts or prepositional phrases can be appended to the requirement. Intransitive requirements consist of much of the same information excluding the object.

<functional requirement>::=

<subject><modal><intransitive verb> {<adjunct>}

The <vehicle> <shall> not <overheat> {<while driving 120 mph for 5 hours>}

<functional requirement>::=

<subject><modal><transitive verb><direct object> {<adjunct>}

The <indicator light> <shall> <alert> the <crew> {<when the vehicle is in reverse>}

Non functional requirements describe characteristics of an artifact and are linking verb requirement sentences. To adequately convey the characteristics, non functional

requirements should possess all of the essential elements of linking verb NL sentence. This includes an artifact, necessity verb, linking verb, and a complement.

<non functional requirement>::=

<subject><modal><linking verb><complement> | {<adjunct>}

The <gaskets> <must> <be> <leak proof> | {<in all orientations>}

These four components allow the reader to know what artifact the requirement is affecting, the level of importance of the requirement, and the characteristic the artifact should possess. In addition to the four essential components, the reader now knows to what extent by the addition of the adjunct.

#### 4.3.2 Specificity

The specificity of an engineering requirement describes the amount of detail about a behavior or characteristic of a system or system component. Adjuncts are used to quantify how much detail is in a requirements statement. A requirement with no adjuncts would be an abstract requirement and does not restrict design space much. A requirement with a large number of adjuncts could be too specific and could narrow down the solution space too much. Quantifying the level of specificity of an engineering requirement is important because it could indirectly help to flag over specified and ambiguous requirement statements.

Determining how qualitative a requirement statement is helps to indirectly identify requirements that are under specified. By replacing weak phrases such as,

minimize, adequate, and maximize, with quantitative phrases, the specificity of the requirement statement will increase. To have a fully specified requirement each adjunct should pair a quantitative phrase with each adjunct.

By identifying the qualitative nature of engineering requirement statements could help to identify high level requirements. High level requirements are not specific as they do not limit the design space for designers. By identifying the number of qualitative and quantitative requirements document the abstraction level of the requirement document can possibly be determined.



## CHAPTER FIVE: FMTV TEST CASE

To demonstrate the capabilities of the proposed linguistic analysis method, a requirements document was analyzed from the US Army's family of medium tactical vehicles (FMTV). The FMTV is a family of vehicles that share the same general architecture but are tailored to achieve different specialized functions. The FMTV perform general resupply, ammunition resupply, maintenance, recovery, engineer support, serve as weapon system platforms and combat service support units. The requirements for the FMTV were gathered from the technical data package (TDP) that identified specifications at a component level. The requirements were focused on defining the physical and performance characteristics of the FMTV. The TDP provides several different types of information about the system including [25]:

- the overall system design, including subsystems, modules and the interfaces
- specific functional capabilities provided by the system
- performance and design specifications
- design constraints, applicable standards, and compatibility requirements
- personnel, equipment, and facility requirements for system operation, maintenance, and logistical support
- manufacturer practices for assuring system quality during the system's development and subsequent maintenance and
- manufacturer practices for managing the configuration of the system during development and for modifications to the system throughout its life cycle.

From the TDP, requirements were extracted and a spreadsheet was created to organize the FMTV requirements. A sample set of requirements from the FMTV TDP are given in Table 5.1. Due to security and classified information in the FMTV TDP, the detailed requirements are not included. Further, specific values and details have been removed from several example requirement statements and replaced with “XXXX”.

**Table 5.1 Sample set of FMTV requirements**

- Engine must XXXX headed up and down slope.
- A mechanical XXXX between XXXX wheel and axle XXXX mechanisms shall exist under all conditions.
- Trailer brake system shall operate when towed by XXXX equipped XXXX.
- Other FMTV models shall perform to a level appropriate to their XXXX.
- The system shall fill completely, with an XXXX XXXX feature to preclude air cavitation at any XXXX fill XXXX up to the maximum XXXX.
- The dashboard fuel level gage shall operate within a XXXX rate.
- Also, exhaust system mounting brackets and fasteners shall protect against dissimilar XXXX.
- Washer reservoir shall not leak when the cab is XXXX XXXX for maintenance.
- Due to safety concerns, reverse gear shall be at XXXX of test for a minimum length of XXXX.

One hundred and ten (110) individual requirement statements from the FMTV TDP are analyzed. The requirements are strategically chosen so a homogenous set of requirement statements could be analyzed. This ensures an even sample of the multiple types of requirement types represented in the FMTV requirements document. The FMTV requirements were linguistically analyzed based on the following:

- Verb type (Section 3.1.3)
- Missing linguistic elements (Section 3.2)
- Number of Adjuncts (Section 3.1.3)

- Count of quantitative values (Section 4.3.2)
- Whether the linking verb was followed by a noun, participle, or adjective (Section 3.1.2)

A snippet of the requirements and analysis fields are shown in Figure 5.1. The first column consists of the requirement text gathered from the TDP. The second column represents the missing linguistic elements of the requirement statement, which determines completeness. The third and fourth columns represent the count of adjuncts present and the count of numerical values present in the requirement statement. These help to quantify specificity within the requirement statement. The last column identifies the parts of speech of the subject complement in a non-functional requirement. This column represents either a qualitative (adjective) or quantitative (noun) non-functional requirement.

<b>Requirement Text</b>	<b>Completeness</b>	<b>Verb type</b>	<b>Adjunct</b>	<b>Cardinality</b>	<b>Noun Participle or Adjective</b>
Engine must XXXX headed up and down slope.	Full	Intransitive	2	0	N/A
A mechanical XXXX between XXXX wheel and axle XXXX mechanisms shall exist under all conditions.	Full	Intransitive	3	0	N/A
While fording, the engine shall be capable of being restarted when XXXX for XXXX.	Full	Linking	1	1	Adjective
XXXX shall be provided at the XXXX and XXXX of vehicle	Full	Linking	2	0	Adjective
The service brakes shall control and hold the vehicle on a XXXX grade, when XXXX XXXX or XXXX slope.	Full	Transitive	3	1	N/A
Each model shall have a XXXX ratio (MR) no greater than specified in XXXX.	Full	Transitive	1	1	N/A
The suspension design shall limit the XXXX natural frequency of the XXXX to a maximum of XXXX hertz.	Full	Transitive	2	1	N/A

**Figure 5.1 Snippet from FMTV requirements analysis spreadsheet**

The analysis method (Figure 4.1 in Section 4.2) is then applied to each of the individual analyzed requirement statements to determine completeness, functionality, and specificity.

To determine the completeness of all the requirements examined, each statement was examined manually to identify any missing linguistic elements based on the syntax

(Table 4.3). If a linguistic element was missing, the missing element was recorded in the spreadsheet. The results from the completeness column are shown in Figure 5.2.



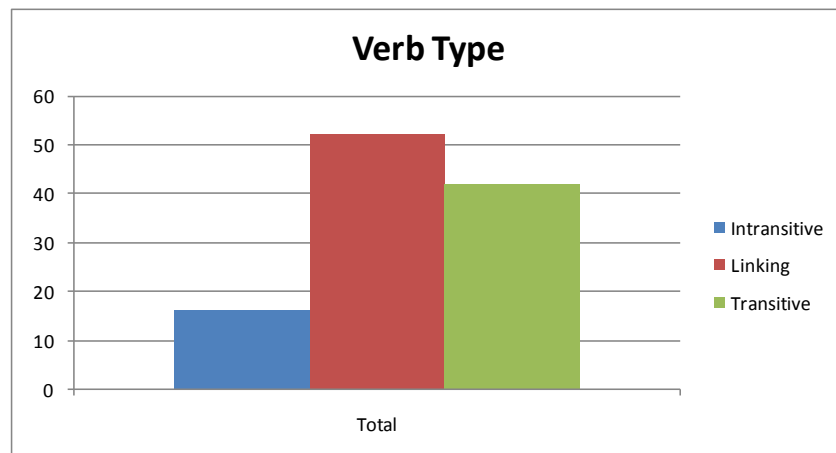
**Figure 5.2 Completeness of FMTV requirements**

Out of the 110 requirements, 105 were complete and 5 were incomplete. This gives the requirements examined a 95% completeness rating. Taking a deeper look at the five incomplete requirements, it was observed that the only missing linguistic element was the subject. This is an expected result because the requirements documents examined was a component level document. Component level documents are the result of multiple iterations on requirement documents that occur earlier in the design process. The multiple iterations help to refine the requirement statements making them more complete. The requirements that were missing the subject are shown in Table 5.2

**Table 5.2 FMTV requirements with missing subject**

Both sections shall be XXXX adjustable.
Being lift towed at XXXX for a distance of at least XXXX miles at a speed of XXXX mph, without preparation except that the XXXX to the wheels in contact with the XXXX shall be XXXX, without XXXX to either XXXX.
Being towed at XXXX by a like vehicle (see paragraph 6.3.14) for a distance of at least XXXX miles at a speed of XXXX mph, with XXXX on ground, without XXXX, without XXXX to either XXXX.
If not equipped with XXXX, shall permit vehicle speeds up to XXXX mph for XXXX on roads, trails, and cross-country.
There shall be no evidence of XXXX to the engine while performing these operations.

The next step in the analysis method is determining whether the requirements were functional or non-functional. To determine the functionality of the requirement statements, the main verb type was identified as transitive, intransitive, or linking in each requirement statement. The results from observing the main verb type is shown in Figure 5.3.



**Figure 5.3 Main verb types**

In the 110 requirements, 15% used intransitive verbs, 38% used transitive verbs, and 47% used linking verbs. Based on the syntax presented in Section 4.1, the requirements

document has 53% functional requirements and 47% non-functional requirements. Now that the functionality of the requirements has been determined, further examination can reveal the specificity of the requirements.

The specificity of requirements is determined by the number of adjuncts, and numerical values. In addition to adjuncts and numerical values, which are used for all requirements, the specificity of non-functional requirements can also be determined by the part of speech in the subject complement. The subject complement of a non-functional requirement can be a noun, adjective or participle. A non-functional requirement where a noun is the subject complement typically leads to quantitative non-functional requirements which can also be classified as form requirements which are shown in the table below.

**Table 5.3 Non-functional requirements where the subject complement is a noun**

The XXXX assembly shall not be more than XXXX inches (XXXX inches for MTV Expansible Van) XXXX of the XXXX part of the vehicle.
Due to safety concerns, reverse XXXX shall be at mid-point of XXXX for a minimum length of XXXX feet.

In addition to nouns, participles also create non-functional requirements. The non-functional requirements that result from the use of participles typically specify that some behavior must take place that is external to the system. In the first requirement in Table 5.4, the air inlet is not locating anything rather the requirement is specifying that something external to the system must locate the air inlet. The other requirements in Table 5.4 have the same characteristics.

**Table 5.4 Non-functional requirements that consist of participle**

The air inlet shall be located to ensure that XXXX entry during XXXX and XXXX shall occur.
Temperatures shall be recorded XXXX feet (XXXX m) above the ground.
All spaces shall be marked with suitable XXXX describing the XXXX to be stowed in the respective XXXX.
Brake linings shall be constructed from XXXX materials.

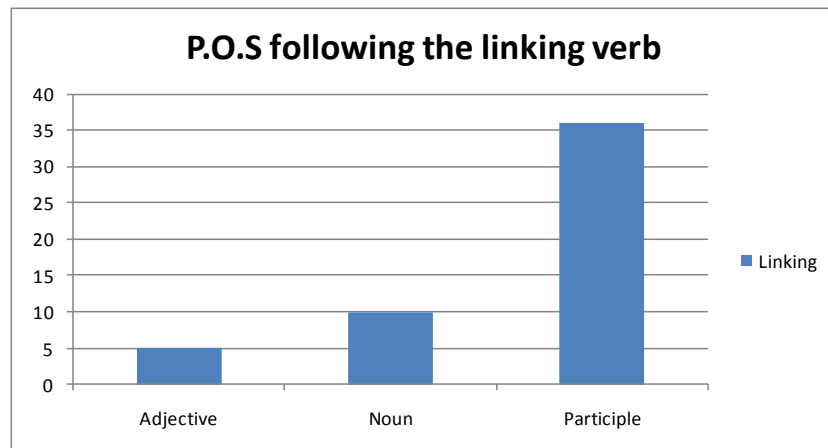
Lastly, the subject complement in a non-functional requirement can be an adjective. These types of requirements tend to be non-functional qualitative requirements. This type of requirement describes characteristics that a system should possess. Requirements of this type are shown in Table 5.5, where in the first requirement the characteristic is being adjustable and in the second requirement the characteristic is for the locations to be accessible.

**Table 5.5 Non-functional requirement where the subject complement is an adjective**

Both sections shall be separately adjustable.
All selected locations for equipment storage, shall be readily accessible to the crew.

Of the 51 linking verb requirements, 71% contained a participle, 23% contained a noun and 6% contained an adjective as the subject complement which is shown in Figure 5.4.



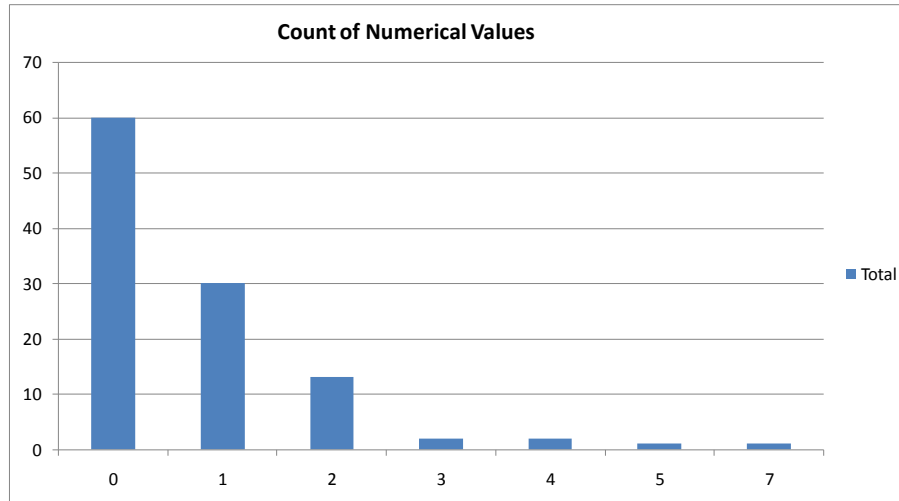


**Figure 5.4 Breakdown of parts of speech following the linking verb**

The low percentage of non-functional qualitative requirements is a result of the requirements documents being a component level document, as most qualitative requirements occur in early system level documents [15]. The high percentage of non-functional requirements that specify a function external to the system is also a result of the document being at the component level. These requirements typically are specifying the geometry or assembly of the components in the system. However, more form requirements would be expected in a component level document that was observed in this study.

Next in determining the specificity of a requirement is examining the count of numerical values. Numerical values within a requirement statement add a quantitative component to the requirement statement which makes the requirement more specific and less ambiguous. In the document numerical values were considered as numbers that represented a complete set. For example, in a requirement statement that contained 0 – 60, the count of numerical values would be one not two, as 0-60 represents a complete

set. The breakdown of the number of numerical values present in the requirements examined is shown in Figure 5.5.



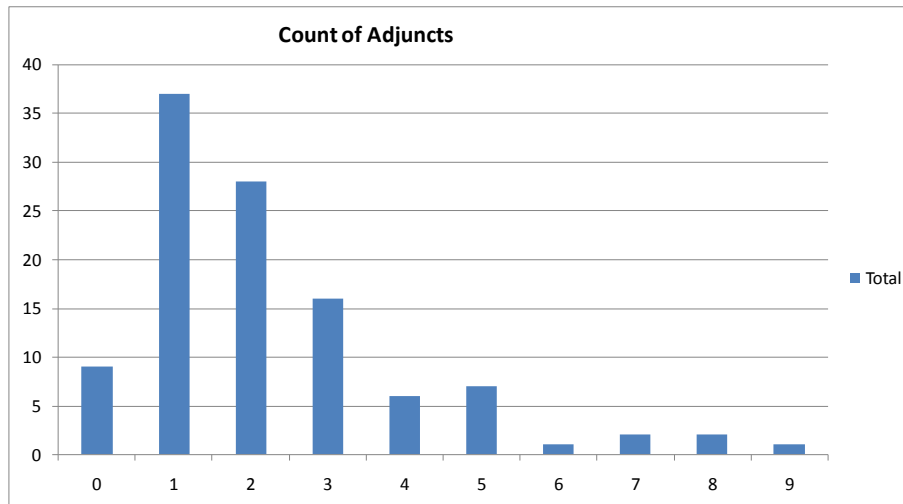
**Figure 5.5** Count of numerical values present in the examined requirements

Of the 110 requirements examined 55% of them contained no numerical values. Many of these requirements can be made more specific and less ambiguous by adding numerical values. These values would further enhance the designer’s capabilities of completely understanding the intent of the requirement statement. Although numerical values tend to decrease ambiguity within a requirement statement, it can also narrow the solution space down too much. This is a negative if there is no traceable justification for the numerical value. In the second requirement in Table 5.6, a numerical value would be appropriate to replace the ambiguous phrase “highest grade” as this will differ from designers. However, in the third requirement a numerical is not necessarily needed, because the characteristic “waterproof” encompasses the quantitative component.

**Table 5.6 Examples of requirements with no numerical values**

The system shall fill completely, with an XXXX feature to preclude air XXXX at any coolant fill rate up to the maximum fill rate.
Workmanship shall be of the XXXX consistent with the intention of this specification.
Seams shall be coated with a sealer to provide a XXXX joint.
An indicator light shall XXXX the crew when the parking brake is engaged.

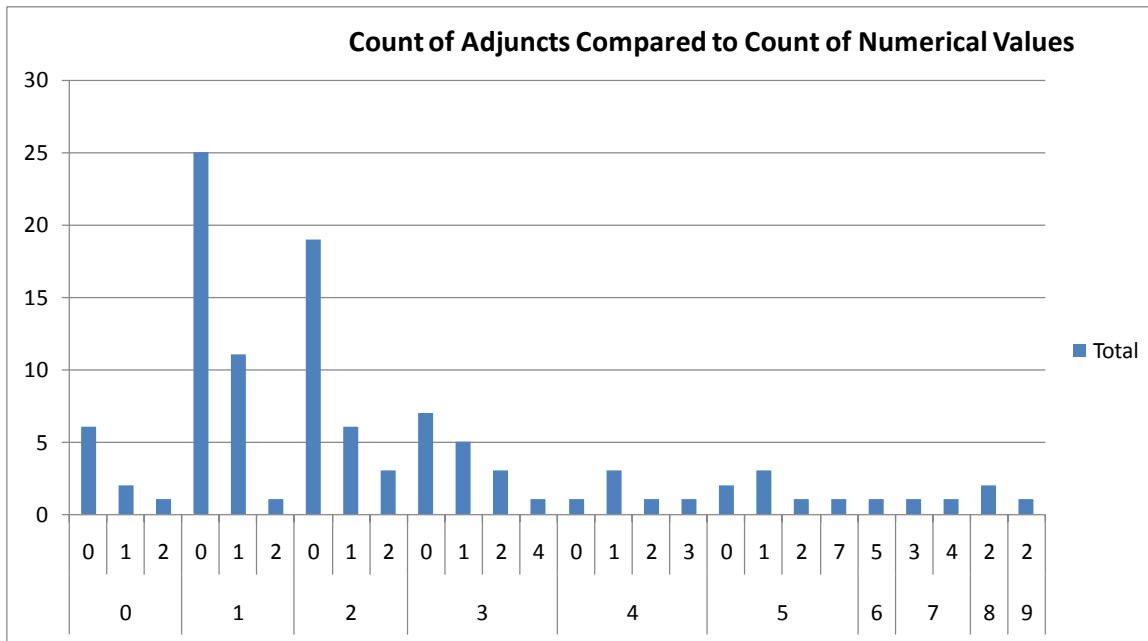
Next in determining the specificity of a requirement statement is to examine the count of adjuncts. As stated in Section 3.1.3, adjuncts are phrases that add additional information to a NL sentence. The additional information added the requirement statement creates a more specific requirement. Figure 5.6 shows the distribution of the number of adjuncts present in the examined requirement statements.



**Figure 5.6 Count of adjuncts of examined requirement statements**

For the requirements examined 92% of them contained at least one adjunct. This is to be expected as the document examined was a later stage component level requirements document.

To further understand the level of specificity of a requirement, the count of adjuncts should be compared to the count of numerical values in a requirement statement. The number of numerical values should equal or exceed the number of adjuncts for a requirement to have a high level of specificity. Equal numerical values and adjuncts mean that for every additional piece of information concerning the requirement, there is a quantitative value associated with it to explicitly express to what extent to the user. This is demonstrated in Figure 5.7



**Figure 5.7 Adjuncts compared to numerical values in examined requirements**

Of the requirements examined 15% had equal or a greater number of numerical values compared to the number of adjuncts. This low percentage provides opportunity for the requirements document to be further refined increasing the specificity and reducing the ambiguity of the document.

The linguistic analysis of the examined requirements reveals that:

- The requirements contained in the documents are mostly complete
- The document is comprised of half functional and half non-functional
- Majority of the non-functional requirements describe actions that are to be performed on the system not by the system (i.e. assembly requirements)
- There is opportunity for improving the specificity of the requirement statements in the document

## CHAPTER SIX: CONCLUSION AND FUTURE WORK

Based on the syntax and the analysis method for modeling engineering requirements, presented in Chapter 4, and demonstrated on 110 requirements for the Family of Medium Tactical Vehicles (FMTV), the research questions formulated in Chapter 1 are answered. Further, several conclusions and future work are identified from these answers.

### **6.1 Answers to Research Questions**

Research Question 1: What is a structure or set of structures for expressing engineering requirements based on linguistics and grammar?

Answer: The linguistic structures for expressing engineering requirements are shown in Table 4.2, Table 4.3, and Table 4.4. The general linguistic structure of an engineering requirement statement consists of a subject, modal, main verb, and subject complement, object, or adjunct. The subject represents the artifact being designed, the modal shows necessity within the statement, the main verb displays behavior or allows for characteristics to be linked, and the subject complement, object, or adjunct represent the conditions of the statement.

As stated in Section 1.1, engineering requirements are statements that adhere to the same grammatical rules and construct as any other NL sentence. Therefore a three-tiered linguistic and grammatical approach was taken to determine the structure for

expressing engineering requirement statements. The three tiered approach consisted of analyzing the parts of speech (POS), sentence structure, and main verb type to determine a formalized requirement statement structure. Parts of speech tagging is used to identify the functionality of all the words in a requirement statement. Once the parts of speech have been identified for all the words in the requirement statement, the relationship between the words must be identified. This is achieved by examining the sentence structure of the requirement statement. Examining the sentence structure adds context and adds semantics to the tagged words. Once the relationships between the words in the requirement statement have been established the main verb in the requirement statement is classified as intransitive, transitive, or linking. This allows the requirement statement to be classified as functional or non-functional. The three-tiered approach leads to a syntax for functional and non-functional requirements based on parts of speech, sentence structure and main verb type.

Research Question 2: What types of analysis on engineering requirements can be completed using the formalized structure developed from linguistics and grammar?

Answer: Using the formalized structure for expressing engineering requirement statements, an analysis procedure was developed to determine completeness, functionality, qualitativness, quantitativness, and level of specificity of an engineering requirement statement

The analysis process used to analyze engineering requirement statements is shown in Figure 4.1. The first step in the process is to ensure that a complete requirement is being processed upon. This is accomplished by using the syntax developed in Chapter 4, to ensure that all the essential elements of a requirement statement are present. Once the requirement statement is deemed complete, the main verb can be identified as transitive, intransitive or linking. This determines whether the requirement statement is functional or non-functional. This step is important because depending on whether the requirement is functional or non-functional will affect how the requirement statement is further analyzed for specificity.

Once the functionality of the requirement statement is determined, the level of specificity and whether the requirement is qualitative or quantitative can be determined. Determining whether functional requirements are qualitative or quantitative is based on the presence of numerical values in the object or adjuncts of the requirement statement. Determining whether non-functional requirements are qualitative or quantitative is based on the parts of speech contained in the subject complement. If a noun is present in the subject complement the non-functional requirement is most likely quantitative. If an adjective is present in the subject complement the non-functional requirement is most likely qualitative.

The next step in the analysis process is determining the level of specificity of an engineering requirement statement. This is determined by identifying the count of numerical values and adjuncts in a requirement statement. Numerical values present in the condition of a requirement, has the possibility to reduce ambiguity and increase



specificity within a requirement statement, and adjuncts further constrains the solution space by adding additional information to the statement. Furthermore, the count of numerical values can be compared with the count of adjuncts to determine the complete specificity of a requirement statement. Ideally, there should be an equal or greater count of numerical values compared to adjuncts.

## **6.2 Contribution**

Two key contributions results from this research. The first is a formalized syntax for representing engineering requirements based on linguistic and grammatical elements. This syntax uses parts of speech, sentence structure, and verb classification. Furthermore, a complementary method to analyze the requirements has been proposed in this research. The formalized syntax is derived from linguistics and grammar from the English language. This allows a requirement to be decomposed into the following linguistic and grammatical constituents: subject, modal, main verb type, and verb phrase. These linguistic elements are then translated into terms that better relate to engineering requirements which are system, necessity, behavior/function, and condition. The formalized syntax leads to more completely expressed engineering requirement statements. Requirement specificity can be objectively determined by analyzing phrases that add additional information to the requirement along with the pairing of quantitative values.

The formalized syntax presented in this research leads to the development of a method to analyze engineering requirement statements. The linguistic elements allow for the individual parts of an engineering requirement statement to be analyzed rather than it

being analyzed as a single entity. The analysis method proposed allows for engineering requirement statements to be mostly objectively categorized with some subjectivity occurring due to the minimum amount of semantics captured. The categories that engineering requirement statements can be classified as are, functional, non-functional, qualitative and quantitative. In addition to these categories, the analysis method also enables a requirement's level of specificity to be quantified as demonstrated within the FMTV case study.

The limitations of the syntax and analysis techniques relate to the ill defined nature of the English language. The first limitation of the proposed syntax and analysis method, concerns using intransitive, transitive, and linking verbs, to objectively define whether the requirement is functional or non-functional. Intransitive and transitive are defined as being action verbs; however these actions may not always be directly translated to engineering actions. For example have, include, and possess display the action of possessing something. In the example below the requirement is describing the intent of the system to possess an object. This is describing how not what which makes this requirement a specification.

*The system must include an object*

In the domain of engineering requirements, these verbs show no engineering action; instead they define a solution or specification for a system. The syntax and analysis techniques can only handle ambiguity that is caused by poor sentence structure or

incompleteness. Finally, the proposed method does not possess the ability to identify semantically ambiguous requirement statements. Semantically ambiguous requirements deal more with the human reasoning aspect of deciphering an engineering requirement statement.

### **6.3 Future Work**

The analysis method discussed in this thesis only addresses two quality attributes of requirement statements, incompleteness, and specificity. To increase the use of the syntax, analysis methods that address other quality properties such as ambiguity, understandability, and traceability needs to be directly addressed. Fully addressing ambiguity would require capturing the complete semantics of a sentence. Capturing all the semantics of a requirement statement would likely lead to the objective interpretation of every aspect of a requirement statement.

Traceability of engineering requirement statements would require the hierarchical component of a requirements document to be captured. This would possibly allow for solution specific requirements to be identified. The abstraction level of the requirement statements could also possibly be determined if traceability could be addressed. The abstraction level of a requirement statement could possibly indicate how much of the design has been completed. For example, if the document contains a large percentage of high level requirements that would indicate the beginning stages of design. If a requirement documents contains a large percentage of requirements using possessive verbs, requirements with high numbers of adjuncts and numerical values that could

possible indicate a design that is reaching completion. This could also have implications in using the requirements document as a timeline for the design process.

To ensure the methods developed in this thesis decreased the amount of incomplete requirements, increased the author's ability to objectively classify functional, non-functional, qualitative and quantitative requirements, and correctly determine the specificity of a requirement statement, they should be rigorously compared with other methods such QuARS and the ARM tool. Also a comparison of the proposed method to methods that use empirical information to improve the expressiveness of requirement statements would prove that a linguistic rule based approach would better enhance the expressiveness of requirement statements. In addition to comparing the methods developed in this thesis to other methods, a user study should be conducted to capture whether the requirements classifications are completely objective.

To increase the usability of the methods discussed in this thesis, the proposed analysis techniques should be implemented programmatically. This would include a graphical user interface and database that would force users to author complete requirements from the beginning. A database would be needed to store the complete requirements for analysis to determine functionality, whether the requirements are qualitative or quantitative, and level of specificity of the requirement statements. Programmatically determining these characteristics would make using the tool more practical.

## APPENDICES

## Appendix A: PennTree Bank Tag Set [26]

1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential <i>there</i>
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NP	Proper noun, singular
15.	NPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PP	Personal pronoun
19.	PP\$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	<i>to</i>
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP\$	Possessive wh-pronoun
36.	WRB	Wh-adverb

## REFERENCES

- [1] J. O. Grady, *System verification: proving the design solutions satisfied the requirements*. New York, New York: Elsevier, 2007.
- [2] G. Pahl, W. Beitz, and H. Grote, *Engineering design: a systematic approach*, 3rd ed., K. Wallace and L. Blessing, Eds. New York: Springer, 2007.
- [3] G. Pahl, W. Beitz, J. Feldhusen, and K. H. Grote, *Engineering Design: A systematic Approach*. London, UK, 2007.
- [4] A. Lamsweerde, "Requirements engineering in the year 00: a research perspective," in *International Conference on Software Engineering*, Limerick, Ireland, 2000, pp. 5-19.
- [5] I. Hooks, "Writing Good Requirements," in *Fourth International Symposium of the NCOSE*, 1994, pp. 197-203.
- [6] E. Hull, K. Jackson, and J. Dick, *Requirements Engineering*, 2nd ed. 2005: Springer.
- [7] G. Lami, "QuARS: A Tool for Analyzing Requirements," Carnegie Mellon, Pittsburgh, Technical Report 2005-TR-014, 2005.
- [8] "OMG Systems Modeling Language (OMG SysML)," 2008.
- [9] U.S. Department of Defense (DOD), "MIL-STD-961D: Department of Defense Standard Practice for Defense Specification," 1988.
- [10] W. M. Wilson, L. H. Rosenberg, and L. E. Hyatt, "Automated Quality Analysis of

Natural Language Requirement Specifications," NASA Software Assurance Technology Center,.

- [11] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "The Linguistic Approach to the Natural Language Requirements Quality: Benefit of the use of an Automatic Tool," in *Software Engineering Workshop, 26th Annual NASA*, Greenbelt, 2001, pp. 97-105.
- [12] M. Osborne and C. K. MacNish, "Processing natural language software requirement specifications," in *Requirements Engineering. 1996., Proceedings of the Second International Conference on*, Colorado Springs, CO, 15-18 Apr 1996, pp. 229-236.
- [13] Institute of Electrical and Electronics Engineers, "IEEE Standard Glossary of Software Engineering Terminology," New York, IEEE Standard 610.12-1990 1983.
- [14] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: a roadmap," in *International Conference on Software Engineering*, Limerick, Ireland, 2000, pp. 35-46.
- [15] A. Ryan, "An Approach to Quantitative Non-Functional Requirements in Software Development," in *European Conference on Systems Engineering*, Germany.
- [16] "IEEE guide to software requirements specification," *IEEE Std 830 - 1984*, Feb 1984.
- [17] K. Ulrich and S. Eppinger, *Product Design and Development*, 4th ed. New York, US: McGraw-Hill, 2008.
- [18] M. Blaha and J. Rumbaugh, *Object-Oriented Modeling and Design with UML*, 2nd



- ed. Upper Saddle River, US: Prentice Hall, 2005.
- [19] M. Hause, "The Systems Modeling Language - SysML," in *EuSEC*, 2006.
- [20] University of Chicago Press Staff, *The Chicago Manual of Style*, 15th ed. Chicago, US: University of Chicago Press, 2003.
- [21] W. N. Francis and H. Kucera, "Brown Corpus Manual," Department of Linguistics Brown University, Providence, Rhode Island, Manual of Information 1964.
- [22] K. Toutanova and C. D. Manning, "Enriching the Knowledge Sources Used in Maximum Entropy Part of Speech Tagger," in *SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Stanford, 2000, pp. 63-70.
- [23] K. Brown and J. Miller, *Syntax: A Linguistic Introduction To Sentence Structure*, 2nd ed. New York, United States: Routledge, 1991.
- [24] R. S. Scowen, "Extended BNF - A Generic Base Standard," in *Software Engineering Standards Symposium (SESS '93)*, Brighton, UK, 1993.
- [25] "Detail Specifications, Technical Data Packages," MIL-DTL-31000c, 2004.
- [26] M. Marcus, Santorini B., and M. Marcinkiewicz, "Building a Large Annotated Copus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, pp. 313-330, June 1993.
- [27] M. G. Christel and K. Kang, "Issues in Requirements Elicitation," Carnegie Mellon University, Pittsburgh, Technical Report 1992.