5-2010

A Proposed Method to Identify Requirements Significant to Mass Reduction

James Mclellan
Clemson University, jammclellan@yahoo.com

A PROPOSED METHOD TO IDENTIFY REQUIREMENTS SIGNIFICANT TO MASS REDUCTION

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

James Michael McLellan

May 2010

Accepted by:

Dr. Gregory M. Mocko, Committee Chair

Dr. Joshua D. Summers

Dr. Georges Fadel

DISTRIBUTION STATEMENT C: DISTIRBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND THEIR CONTRACTORS. REASON: ADMINISTRATIVE OR OPERATIONAL USE. DATE OF DETERMINATION: 03-02-06. OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO THE U.S. ARMY TANK-AUTOMOTIVE AND ARMAMENTS COMMAND, ATTN: SFAE-CSS-MT, WARREN, MICHIGAN 48397-5000

EXPORT CONTROL WARNING: WARNING – THIS DOCUMENT CONTAINS TECHNICAL DATA WHOSE EXPORT IS RESTRICTED BY THE ARMS EXPORT CONTROL ACT (TITLE 22, U.S.C., SEC 2751, ET_SEQ_) OR THE EXPORT ADMINISTRATION ACT OF 1979, AS AMENDED, TITLE 50 , U.S.C., APP. 2410, ET_SEQ_ VIOLATIONS OF THESE EXPORT LAWS AR SUBJECT TO SEVERE CRIMINAL PENALTIES, DISSEMINATE IN ACCORDANCE WITH PROVISIONS OF DOD DIRECTIVE 5230.25.

ABSTRACT

Reducing the mass of engineering products holds the potential for significant benefits by reducing material costs, environmental impact, transportation costs, and in the case of vehicles, reducing fuel consumption. While there are many approaches for reducing mass, analyzing requirements has the greatest potential since requirements definition is the earliest phase of product development, where the most design freedom exists. This thesis proposes a requirement analysis method that identifies requirements that impact significant amounts of mass. The research hypothesis is: Engineering requirements can be represented and processed in a systematic manner and linked to physical components and systems, thus enabling mass reduction in reverse engineering and product redesign. The approach proposed in this research follows. Engineering requirements are linked to mass through the creation of a standard requirement statement using pre-processing rules and syntax rules. These rules and guidelines are applicable to authoring new requirements and analyzing existing requirements documentation. The processed engineering requirements are linked to physical components and assemblies based on how the requirements affect the components. These relationships are captured in Design Structure Matrices (DSMs) and Domain Mapping Matrices (DMMs). These DMMs and DSMs are used to attain the amount of mass each requirement affects and the level of coupling of each requirement. Further, representations of the requirements, components, and associated relationships are represented using two software tools. First, a systems engineering tool is used to model the system. Second, this model is exported to a traditional spreadsheet application to perform basic mathematical and data filtering functions. Finally, the method is demonstrated on three subsystems of Family of Medium Tactical Vehicle (FMTV) truck.

DEDICATION

This thesis is dedicated to researchers in the field of engineering design. Hopefully this is another step in the right direction leading to better designs. This thesis is also dedicated to my parents. I could never have gotten this far without your help. Thank you for investing your time in me.

ACKNOWLEDGEMENTS

I first of all thank God, the Supreme Designer, for giving me this opportunity to explore more of His world. I secondly would like to thank Dr. Gregory Mocko for all of his insight and knowledge and allowing me to study under him and for recruiting me to work on the ARC project. Thank you for teaching me how to teach myself, and for correcting my southern grammar. I am grateful for the financial support that I have received through my education at Clemson University. I acknowledge funding support from the U.S. Army Tank Automotive and Armament Command (TACOM) through the Automotive Research Center (ARC) research grant W56HZV-04-2-0001. Support also came from the National Science Foundation (Grant No. 0826441). I would also like to thank Dr. Joshua Summers and Dr. Georges Fadel for being my other committee members and for their advice and encouragement throughout this thesis. Thank you for challenging my thinking. I am also grateful to Jonathan R. A. Maier for working with me on the ARC project and for his help to publish papers. I am very thankful for the members of the CEDAR group for their constant encouragement, challenges and companionship. You guys are great.

Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the Automotive Research Center or the National Science Foundation.

List of Publications

Maier, A., J. R., McLellan, M., J., Mocko, G., Fadel, M., G., Brudnak, and M., (2009), "Lightweight Engineering of Military Vehicles through Requirements Analysis and Function Integration", *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, San Diego, CA,

McLellan, J. M., Maier, J. R. A., Fadel, G. M., and Mocko, G., (2009), "A Method for Identifying Requirements Critical to Mass Reduction", *International Design Structure Matrix Conference*, Greenville, SC.

McLellan, J. M., Maier, J. R. A., Fadel, G. M., and Mocko, G., (2009), "Generating Dsm and Dmm Matrices Using Sysml", *Design Structure Matrix Conference*, Greenville, SC.

Mclellan, J. M. , Morkos, B., Mocko, G., Summers, J. D., (2010), “Requirement Modeling Systems for Mechanical Design: a Systematic Method for Evaluating Requirement Management Tools and Languages”, *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Montreal, Canada.

Table of Content

List of Tables

List of Figures

Chapter 1. Introduction to the Problem

The United States Army actively uses approximately 250,000 light, medium and heavy trucks and also 110,000 trailers at home and in theaters around the globe [1]. The cost of fuel for the Army is roughly $13 per gallon in peace-time and between $100-$400 per gallon in wartime to areas that lack established fuel routes [2]. Nygren and colleagues discuss a future time when the supply of oil will not meet the level of demand [3]. According to Hirsch et al, the conservative estimates for the world oil production peak has already passed (2006,2007) while the most optimistic ones put the peak later than 2025 [4]. This information is displayed in Table 1.

Table 1: Projections of the Peaking of World Oil Production [4]

Projected Date	*Source of Projection*	*Background & Reference*
2006–2007	Bakhitari, A. M. S.	Iranian oil executive
	Simmons, M. R.	Investment banker
After 2007	Skrebowski, C.	Petroleum journal editor
Before 2009	Deffeyes, K. S.	Oil company geologist (ret.)
Before 2010	Goodstein, D.	Vice Provost, Cal Tech
Around 2010	Campbell, C. J.	Oil company geologist (ret.)
After 2010	World Energy Council	Nongovernmental org.
	Laherrere, J.	Oil company geologist (ret.)
2016	EIA nominal case	DOE analysis/information
After 2020	CERA	Energy consultants
2025 or later	Shell	Major oil company
No visible peak	Lynch, M. C.	Energy economist

To deal with this challenge, the Army is seeking ways to increase the fuel efficiency of Army vehicles. In a similar direction to Nygren and colleagues, another report was conducted on strategic responsiveness of US armed forces called "Revolution of Military Logistics" and calls for improvements in broad areas of automation, communications, business practices, command

and control relationships and distribution technologies [5]. According to this report, one specific aspect of interest that needs to be developed is rapid distribution technologies. These reports are significant since both reports point out problems that can be relieved by mass reduction. By reducing the mass of the vehicle, the fuel efficiency of the vehicle will increase. Thus the resulting research question formulated and subsequently addressed in this research is:

> How can requirements be related to mass in the early part of the design process, in the design specification (requirements) phase?

To answer this question in the affirmative, requirements will have to be related to mass. The research challenge addressed in this thesis is to create a process for consistently relating requirements to mass. This is accomplished by the following steps.

1. Create pre-processing and requirement syntax rules for stating requirements
2. Create rules for relating requirements to each other
3. Use relational matrices for showing requirement interactions with each other and the system architecture
4. Create rules for relating requirements to components and to themselves
5. Identify a requirement software to implement the proposed method by examining two software with a proposed metric

To show that these research questions have been satisfied, the method is applied to three Family of Medium Tactical Vehicle (FMTV) example problems and validated against the research questions. The method is validated using the validation square approach. Validity is defined as consistency within the method by use of logical induction and/or deduction [6]. The validation square will be used to prove that the method is indeed valid. An illustration of the validation square is seen in Figure 1.

(1) and (2) Theoretical and Structural Validity	(6) Theoretical Performance Validity
(3) Empirical Structural Validity	(4) and (5) Empirical Performance Validity

Figure 1: Illustration of the validation square [6]

Part (1) involves accepting the validity of the constructs used in the proposed requirement analysis method [6]. This is accomplished using an extensive literature review that is included in Chapter 2. The four primary constructs used in this method are requirement capabilities, relational matrices, requirement rules and requirement syntax rules. The two constructs requirement capabilities and relational matrices are well known and have extensive literature discussing them. These will be shown in the thesis when the topics are introduced. Requirement capabilities are discussed in Chapter 2 while relational matrices are discussed in Chapter 3.

Method consistency is the focus of **Part (2)**. To accomplish this, Pederson et al. encourages the use of flowcharts to show the information flow within a method [6]. This is accomplished in Chapter 3 when the proposed requirement analysis method is introduced. This flowchart will show each consecutive step of the method. Proving that the example problems used are acceptable and like other problems that the method would encounter are discussed in **part (3)** [6]. The example problems used will be shown in Chapter 6, Chapter 7 and Chapter 8 with three subsystem examples. The outcome of the method is shown in **part (4)** to prove that the results attained do indeed answer the research questions that were started with [6]. This will be further discussed along with the results in the conclusion in Chapter 9. **Part (5)** involves showing the usefulness of the method [6]. This is shown by explaining how each part of the method significantly contributes to the results attained from the method. This will be shown throughout Chapter 3 in the introduction and discussion of the method. **Part (6)** involves showing the usefulness of the method beyond the example problems [6]. This will be discussed in the conclusion.

Chapter 2. Literature Survey

To date, mass reduction has mainly been accomplished using structural optimization, a mass reducing approach that focuses on altering geometry properties of the design [ref]. This is accomplished in the latter stages of design where the components have been roughly designed and mass is taken away in areas where it is not needed. At these latter stages of design, the design field has become quite limited due to selections of component types and geometries. The changing size of the design field as design progresses is illustrated in Figure 2. Thus, mass reduction is quite restricted at this point in the design. To begin the mass reduction process earlier in the design would greatly increase the freedom with which to reduce mass.

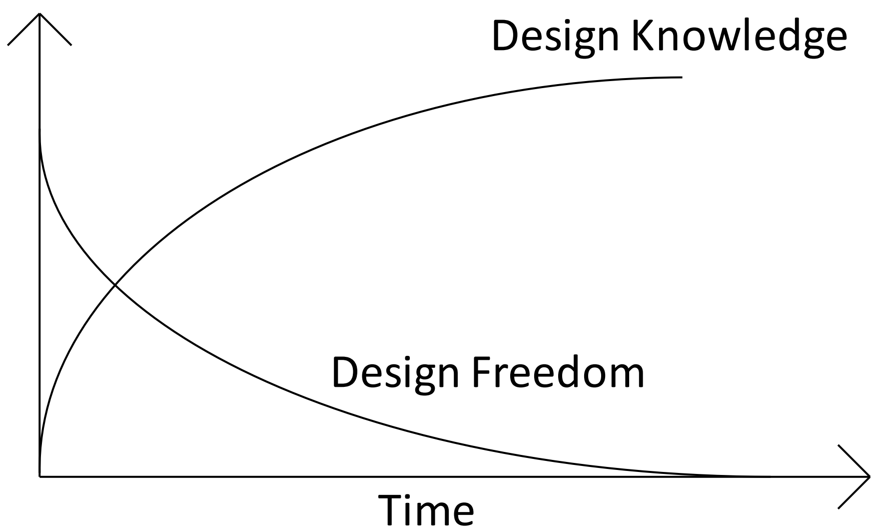


Figure 2: Illustration of the size of the design space during the design process

The requirement phase of the design process is seen by many as the beginning steps in the design method [7-10]. A model of the design process is shown in Figure 3 illustrating the requirement gathering (or Design Specification) as an early phase of the design process.

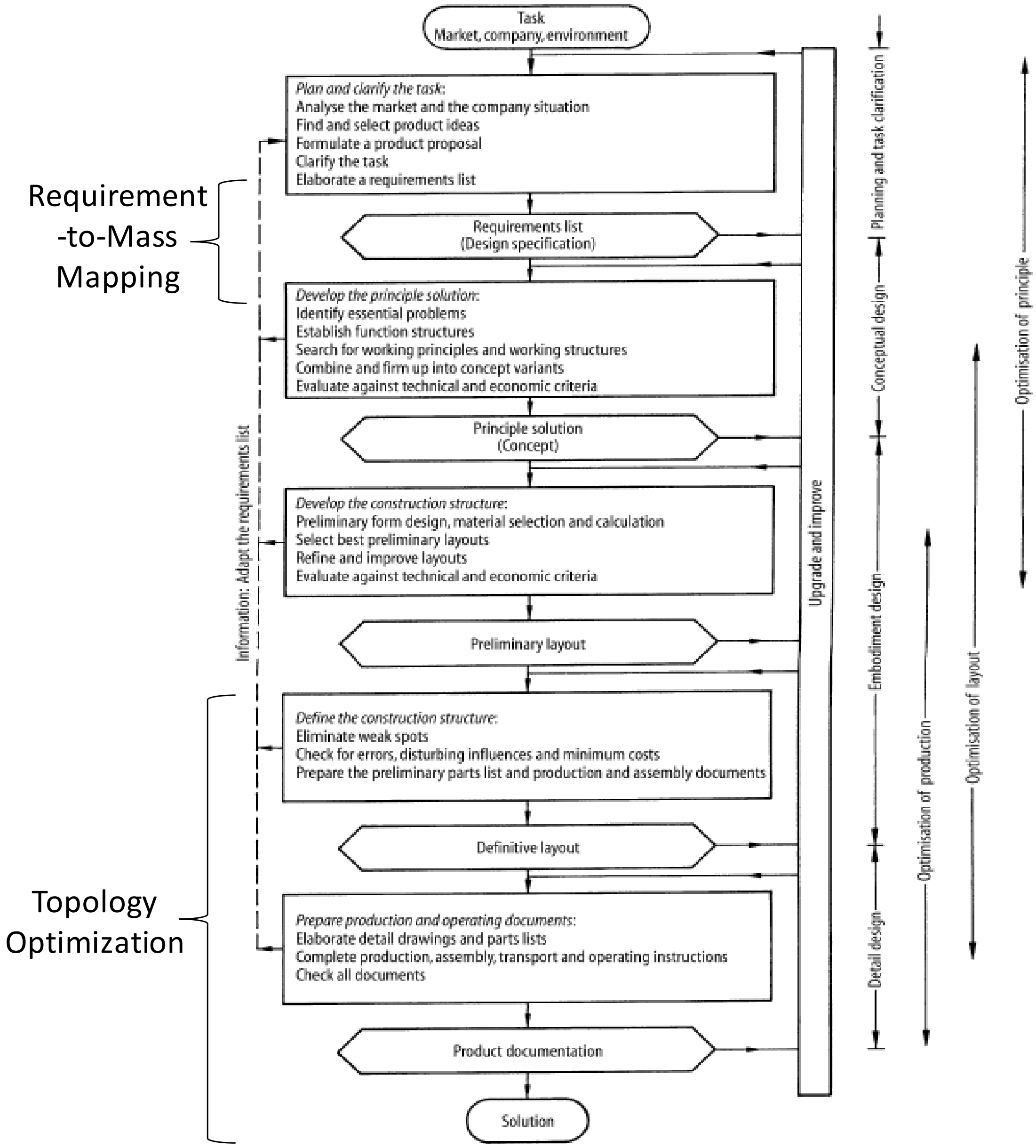


Figure 3: Illustration of the design process [7]

The topology optimization is involved during the middle/end of the design stage.

Requirements (or as some call design specifications) are defined as the goals that engineers design a product to meet or perform to [7,11,12]. Because requirement gathering and specification is largely accomplished at the beginning of the design stage (although it still occurs some throughout the design process), they are extremely important to the design as they set the stage or tone of the design. Getting requirements right is a pivotal part of the engineering process. According to Sud and Arthur, 71% of all software development projects result in complete failure with poor requirements management being one of the main causes of product failure [13]. Though this statistic applies to software engineering requirements, the requirement specification phase is used in software and non-software engineering and thus the importance to getting requirements right applies to non-software engineering also. To focus on mass reduction at this early stage of design shows potential to greatly impact the final product. In the following section the properties and capabilities of engineering requirements are discussed.

Requirement Capabilities

Eight key capabilities are identified from a review of existing literature (see Table 2). These capabilities are focused on the representation and processes associated with engineering requirements. A list of capabilities discussed in this chapter are included in Table 2.

Table 2: Engineering requirement capabilities

Capability	Definition	Ref.
Refinement	Create requirements of narrower scope and higher specificity from parent requirements.	[14]
History	Description of a requirement's evolution through the design process.	[15-17]
Satisfaction	Relationship between a requirement and the artifact designed to fulfill the requirement.	[18,19]
Verification	Relationship between a requirement and the test that ensures the requirement has been satisfied.	[20,21]
Coupling	Interrelationships between requirements.	[16,19]
Prioritizing	Importance ranking of a requirement.	[22]
Input Validation	Ensures quality, structured requirements are input into the software.	[23]
View Restriction	Restricts specific users to viewing a subset of the total requirements.	[16,24]

Refinement

The refinement of an engineering requirement captures additional details and specifics of the requirement. For example, the requirement *"the vehicle must be safe"* is refined by what *safe* means through several additional requirements including *"the occupant cannot experience a G-load of more than 3 Gs in a frontal collision"*, *"the vehicle must not crumple in a roll-over "* and *"the vehicle must not explode when hit from behind."* Each of these requirements further define the safety requirement. The refinement relationship exists between engineering requirements only and may result in a hierarchical requirements structure [14]. It is important to note that refinement does not detail a solution. It rather provides more detail and definition to the requirement.

History

Requirement history is the ability to describe and follow the life of a requirement, in both a forwards and backwards direction" [15]. Requirements history enables the changes and rationale for those changes to be captured. For example, it is important to capture and document the changes between an initial requirement stating *"the vehicle must accelerate from 0 to 60 MPH in 10 seconds"* to the next version stating the *"the vehicle must accelerate from 0 to 60 MPH in 15 seconds."* Requirements history may help designers to avoid costly delays when reusing the requirement in a similar project and provides a means to identify legacy requirements. The rationale for creating or modifying requirements is also included in requirement history.

Satisfaction

Requirement satisfaction is the creation of the physical design to meet the requirement specification. This is where the designer must commit to a physical solution. This physical design may be a system/sub-system/component [18,19]. At the systems level, satisfaction is difficult to model because these requirements affect the entire spectrum of physical components and assemblies. Requirement satisfaction is modeled by explicitly mapping a requirement to the physical system(s) that contribute to fulfilling the requirement. For example, the “engine must dissipate heat” is satisfied by the cooling system.

As the design process progresses and more information is generated about the system, this requirement may be mapped through a satisfaction requirement to the water pump, radiator, thermostat and fan. The satisfaction relationship can be used on multiple levels of decomposition from the system level to the component level. However the satisfaction relationship can only relate engineering requirements to physical entities in the system. An example of satisfaction in engineering design is shown in Figure 4.

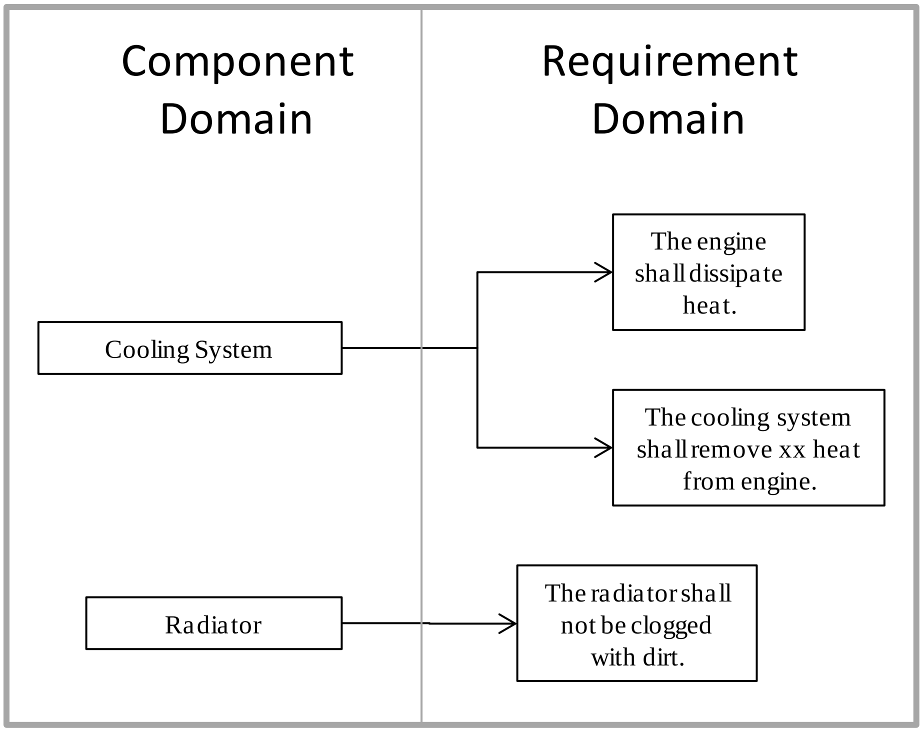


Figure 4: Requirements satisfaction-mapping of components to corresponding requirements

The satisfaction relationship establishes an explicit link between engineering requirements and the physical design (see Figure 4).

Verification

The verification of an engineering requirement indicates how the requirement is tested or evaluated. Requirement verification is often evaluated as *fulfilled/not fulfilled* or *pass/fail* [20]. For example, a requirement on an automotive seat may state *"the seat must be safe during a frontal impact"*. The physical systems that satisfy this requirement may be verified by test procedures that have been established by governmental regulations or industry tests. The verification relationship establishes an explicit link between engineering requirements and testing documentation (see Figure 5).

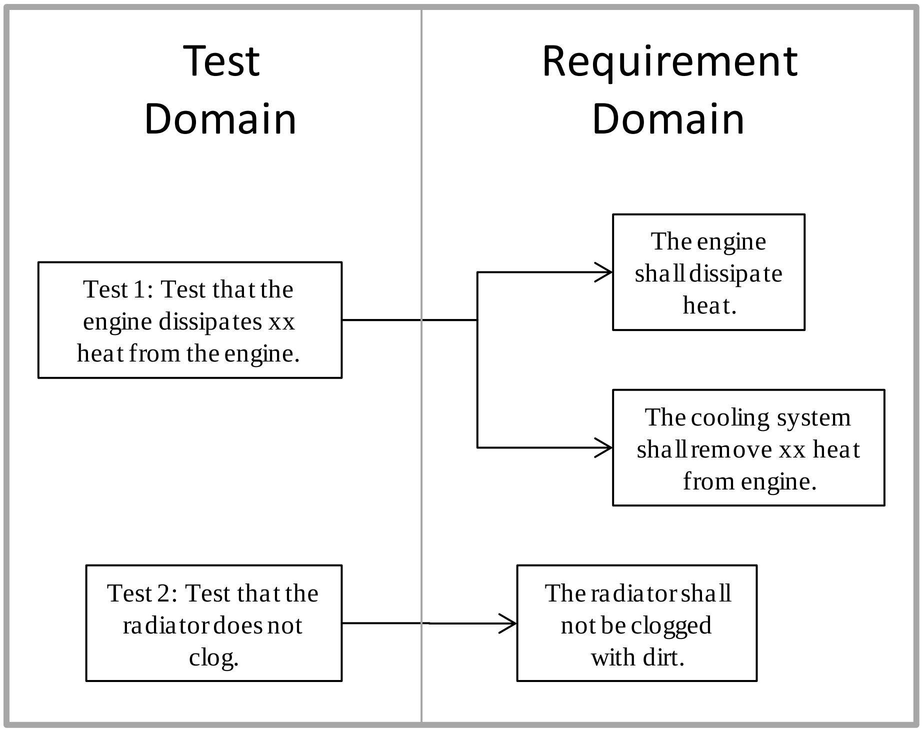


Figure 5: Requirements verification-mapping of tests to corresponding requirements

Coupling

Coupling enables engineering requirements that have an influence on each other to be captured. For example, two requirements are coupled if changing one requirement necessarily changes other requirements [19]. For example, the coupling between engineering requirements is represented using Domain Mapping Matrices (DMMs) (see Table 3) [25].

A 0 in a cell indicates there is no relationship between requirements whereas a 1 indicates a relationship exists between requirements. Coupling can be used to model conflicting requirements. For example, the requirement *"must accelerate from 0 to 60 MPH in 5.5 second"* may be coupled to *"must have a fuel efficiency of 45 MPG."* These requirements are coupled through a physics-based relationship.

Table 3: Requirements coupling-mapping of requirements to requirements

FMTV Component Requirements		A	B	C	D	E	F	G	Total coupling
	A	0	0	0	0	1	0	0	1
	B	0	1	0	1	1	0	1	4
	C	0	0	0	0	0	0	0	0
	D	0	0	0	0	0	0	1	1
	E	0	0	0	0	0	0	0	0
	F	1	1	1	1	1	1	1	7
	G	0	0	0	1	0	0	0	1

Prioritization

Prioritization is used to rank the importance of a requirement [22]. Not all requirements have the same level of importance in a design project. For example in the design of an automotive seat, the requirement "the seat must be safe" may have a greater priority over "the seat must be stain resistant." Prioritizing allows a designer to focus specifically on a select group of requirements to ensure their fulfillment. Current methods for prioritizing include low, medium and high priority levels [26]. Weigers defines a prioritization scale as follows [22].

- Essential- the product must fulfill the requirements.
- Conditional- is not a make-or-break requirement. Is not necessary but would add to the design.
- Optional- functions may or may not be worthwhile

Hull and colleagues identify three types of priority levels [27]. The first two use grammar to define priority levels: Key, Mandatory, Optional and Desirable. The other use Must, Should, Could, Wish (MoSCoW). The third type, importance, uses a numerical grading scale between 1 and 10 [27].

Input Validation

Requirement input validation ensures that quality information is put into the model. Requirement pre-processing rules and syntax rules are potential examples of input validation and can ensure that correct and consistent requirements are used in the model [23]. Requirement pre-processing rules dictate the information to be displayed in the requirement. These include rules on the content of the subject and verb/predicate of the requirement. Once this has been accomplished, syntax rules could be used to ensure uniformity in how the information is displayed. One challenge to providing input validation is the need to create a style, structure and language for the requirements [28].

View Restrictions

Requirement view restrictions filter requirements for different peoples' interests and to minimize design inconsistency [16,24]. An example of view restrictions would be a view reflecting business requirements and a view affecting engineering requirements. Requirements not needed by a certain user only add clutter to the model. Simplification is necessary at this point. The other side of requirement view restrictions is the security side where some requirements are proprietary to specific eyes. Viewing and usage rights are then established based on the user.

Representing requirements

This section will discuss the several ways of representing requirements. Upon a literature review requirement specification can be grouped into three subheadings: Natural Language, Mathematical and Graphical [29,30].

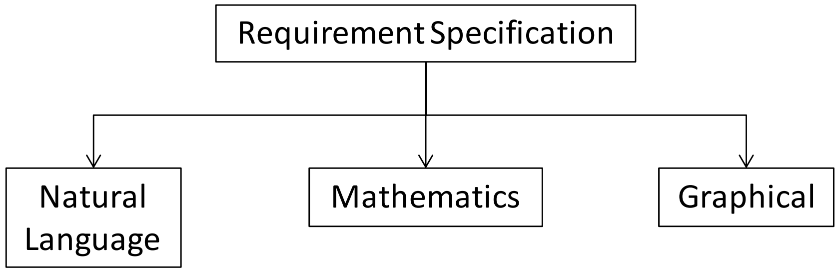


Figure 6: Three approaches to requirement specification: natural language, mathematics and graphical

Natural Language Requirement Representation

Natural language requirements (NLR) are requirements that utilize spoken words to specify the requirement. Natural language requirements are the most flexible since they are written with words and can be phrased to the user's needs [29]. Their flexibility is also a drawback, however. Because natural language is so flexible, it becomes difficult to extract and process information from it in a uniform way. Current methods use either the syntax or the semantics of the sentence. Syntax refers to the organization of words within the sentence. Semantics refers to the meaning of the word itself [31]. Lamar also points out that natural language requirements can lead to ambiguity between customers [32]. To combat these problems of using natural language to specify requirements, Lamar creates a method for determining the correctness of a requirement statement expressed in natural language based on four syntactical elements: artifact, necessity, function and condition. The aspect-oriented requirements engineering (AORE) approach is based on syntactic properties of the statement itself. This has several drawbacks, one of which is that requirement meaning is drawn from the structure of the sentence instead of the semantics of the sentence [33]. Chitchyan et al. propose a different approach called Requirement Description Language (RDL) that uses semantics instead of syntax to model the requirement [33].

Mathematical Requirement Representation

Mathematical specification has the most precision since it uses a numerical method to specify requirements [29]. These, however, have a difficult application and a limited scalability. Z notation (pronounced ‘zed’) is a formal mathematical method for representing the logic used in computer software programs. Formal methods allow computer software (or designs of any type) to be predictable [34]. Usually, requirements are not converted to Z one requirement at a time, but by grouping requirements into a better organized system. Because it is so detailed and precise, Z notation requires extensive training and can only be used by highly trained specialists.

Graphical Requirement Representation

Graphical requirement specification is the most visual type since it models requirements using shapes. One example is Unified Modeling Language (UML) and an extension of UML, Systems Modeling Language (SysML) [24]. An advantage of graphical requirement representation is the ease of which relationships between requirements and components/tests/etc can be created or viewed. UML and SysML software allow relationships to be modeled with ease [35]. A weakness of graphical representation is that it can become clumsy when dealing with large numbers of requirements. It is difficult to find requirements in a diagram if the diagram displays 300 requirements.

This discussion is not meant to be comprehensive but to rather show a sample of different requirement representations. Further, the list is not meant to be mutually exclusive. For example, natural language requirements are often illustrated graphically in languages and tools like SysML

Each approach to requirement specification has its own advantages and drawbacks setting some requirement specification approaches at odds with each other. To address this, work is currently being done to combine approaches such as combining the natural language and graphical approaches. This thesis will combine the use of both (to take advantage of the abilities of both) by using natural language requirements in conjunction with pre-processing and syntax rules to maintain a standardized grammar and graphical requirements to utilize its relation-creating capability.

In this chapter, requirements have been defined and their place in the design process has ben explained. The capabilities of requirements as well as the ways they can be represented have also been discussed. The next chapter will introduce the proposed requirement analysis method to map requirements to mass.

Chapter 3. Mass Reduction Method

Uncoupled Mass Important Requirements Identification Method

Engineering requirements have a great effect on the designed solutions due to their fundamental nature in the design process. In other words, good requirements lead to good designs, bad requirements lead to bad designs. Modifying, adding, or deleting an engineering requirement has the potential to greatly affect vehicle properties For example, adding a requirement that the High Mobility Multipurpose Wheeled Vehicle (HUMMVEE) must be blast resistant to IED and other explosive devices has forced the U.S. military to up-armor these vehicles dramatically affecting their life, fuel consumption and dynamics. To understand and identify how requirements affect mass, a systematic method is required. The method consists of modeling requirements using a formal syntax, verifying if the requirements are stated correctly, mapping the requirements to physical subsystems (i.e., components or assemblies) in the system, and identifying how the requirements affect mass if they are modified, added, or deleted (see Figure 7). This requirement analysis method is accomplished in three steps: identification and modification of requirements, reverse engineering and specific analysis.

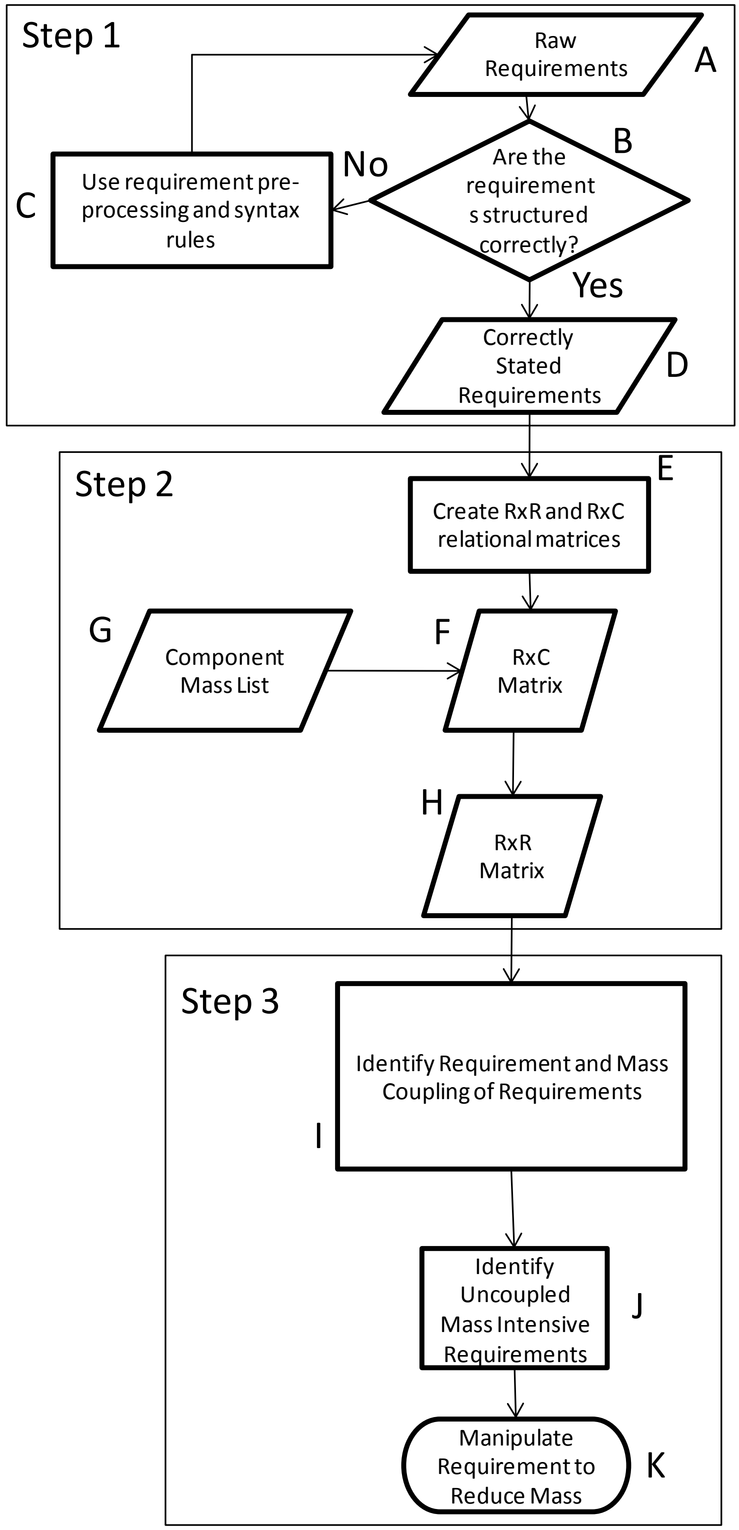


Figure 7: Flowchart of Requirement Analysis Method

In Step 1, the raw requirement list in **Part A** is reviewed to ascertain whether it follows the pre-processing and syntactical requirement rules (**Part B**). Requirement rules are used to standardize the sentence structure to ensure correct format. Parts E of the requirement analysis method depend on the correct sentence structure of the requirement. If the requirements do not follow the preprocessing and syntactical rules, the requirements are reworded to comply with the rules using pre-processing and syntax rules in **Part C** to create the correctly stated requirements in **Part D**. Step 2 involves reverse engineering the design and relating requirements to components and to requirements. The correctly stated requirements are then used to form Requirement vs. Requirement and Requirement vs. Component matrices as in **Part E**. In **Part F**, a Domain Mapping Matrix (DMM) is used to relate Requirements to Components (RxC Matrix) to map the complexity between different design domains. Two types of DMMs are used. The first uses binary relationships (1s and 0s) to describe the relationship between requirements and components. The second type uses a weighted value, the component's mass obtained from the component mass list in **Part G**. Step 3 involves specific analysis of the requirement matrices. The DMMs are used to evaluate component/requirement couplings in **Part I**. A Design Structure Matrix (DSM) in **Part H** is used to relate requirements to each other (RxR Matrix) to describe the coupling between requirements. The DSM matrix is obtained by either:

1. Multiplying the RxC mass matrix with the transpose of the RxC binary matrix
2. Multiplying the RxC binary matrix with the transpose of the RxC mass matrix

Uncoupled requirements in **Part J** as well as the mass intensive requirements from **Part I** are used to find requirements that are both uncoupled and mass intensive in **Part J**. These are then manipulated in **Part K** to reduce mass. A BMW subsystem is used as an example case to implement the method in steps 2 and 3. However, the BMW subsystem lacked requirements that applied to all of the rules discussed in step 1. As examples in step 1, the Family of Medium Tactical Vehicle (FMTV) requirements are used.

Step 1: Acquire and process requirements

Uniform Requirement Statement

The first step in the method is stating each requirement according to a specific syntax and information content. This is done to ensure that all requirements are stated in a uniform format and thus can be processed and analyzed. Ten pre-processing rules are used to ensure that complex requirements are decomposed into simple requirement statements. These rules were developed by identifying the parts of speech within a requirement statement and seeing if and how they were used within the original requirements located in ATPD2131F.1. They were also developed to aid with constructing the DSM and DMM matrices in Step 2 of the requirement analysis method. Some requirements lacked a subject, giving rise to Rule 1. Other requirements had compound subjects and verbs, making it difficult to relate component and function domains to each other. This gave rise to Rules 2 and 3. Other requirements contained clauses located in various places throughout the requirement statement. To standardize the location of these clauses, Rules 4,5,6 and 7 were created. Some requirements were found to include descriptions for how the test to validate that requirement was to be accomplished. This led to defining that the scope of a requirement should be to determine what objective the requirement should accomplish and what properties must if have or not have [7]. This led to the creation of Rule 8. Many requirements included functions of the design but were obscured by the way the requirement was written. Rules 9 and 10 were created in an effort to make the functions of the design obvious. In this context complex refers to a requirement that contains multiple subjects, multiple behaviors, and multiple conditional clauses. The analysis methods proposed in this research is based on the analysis of simple requirements. Syntax rules are then used to ensure the layout of the requirement. The second phase ensures the correct syntax for a requirement statement. These two phases are completed in an iterative manner. The pre-processing rules ensure that the information for the subject, verb and adjective phrase are of a certain type. Each of the pre-processing rules and associated examples are presented below using the Family of Medium Tactical Vehicles (FMTV) requirements. Outside of the scope of this thesis, these requirement rules were created to bring a scope to what the requirement statement must accomplish. These rules are also used to maintain the understandability of the requirement to the reader by requiring the information to be present and in a certain location within the requirement statement.

Pre-processing Rules

Rule 1: *The subject of the requirement must always be a physical or tangible system, subsystem or component and not a property/attribute of a physical artifact*. This is codified due to foresight that requirements should be able to be related to each other according to subjects. Thus, requirements with the same subject are coupled to each other. A requirement with an aspect as a subject would be as follows.

> <u>Fluid line protection</u> shall be ensured by placement near heavier components.

In this requirement, the subject of the requirement is ‘fluid line protection’. Instead of writing an aspect, the subject should be ‘fluid lines’. The correct way to write this requirement would be as follows.

> <u>Fluid lines</u> shall be protected by placement near heavier components.

Rule 2: *Requirements with multiple systems must be decomposed into separate requirement statements*. This is codified to maintain simplicity in requirements. Also, this enables requirements to be related to each other by similar subjects. An example of a requirement with multiple subjects is as follows.

> All vehicle and kit configurations shall not have inherent adverse electrical characteristics.

The compound subject of the requirement is ‘vehicle configurations’ and ‘kit configurations’. This requirement should be decomposed into two separate requirements, one with ‘vehicle configurations’ and the other with ‘kit configurations’.

> All vehicle configurations shall not have inherent adverse electrical characteristics.

> All kit configurations shall not have inherent adverse electrical characteristics.

Rule 3: *Requirements with multiple verbs must be decomposed into separate requirement statements*. Just as the subject of a requirement should contain only one subject, the requirement should contain only one verb to ensure requirement simplicity and be able to relate requirements according to verb type. An example of a requirement with multiple verbs is as follows.

> The rear view mirrors shall be re-adjusting and self-indexing without the use of tools.

In this requirement, the verb consists of two functions, ‘re-adjusting ‘ and ‘self-indexing’. This requirement should be decomposed into two requirements, one with ‘re-adjusting’ as the verb and the other requirement with ‘self-indexing’ as the verb.

> The rear view mirrors shall be re-adjusting without the use of tools.

> The rear view mirrors shall be self-indexing without the use of tools.

Rule 4: *Requirements with exception clauses must be located at the end of the requirement statement*. This rule groups the body of the requirement together, while keeping the exception at the very end. If the exception is located in the middle of the requirement, the attention of the reader is averted from understanding what the requirement is about to what the requirement affects and then redirects the reader again to what the requirement is about. An example of a requirement with an exception clause in the middle is as follows.

> Wiring not protected from accidental contact with troops, terrain, or vegetation <u>unless otherwise specified herein</u> shall be of a large size.

In this requirement, the reader's attention is directed to what the requirement is about, then the attention is transferred to an exception, and finally the reader's attention is once again directed back to what the requirement is about. A better way to phrase this requirement is as follows.

> Wiring not protected from accidental contact with troops, terrain, or vegetation shall be of a large size <u>unless otherwise specified herein</u>.

The reader's attention is drawn to what the requirement is about, and then to the exception of the requirement.

Rule 5: *Requirements with subject description clauses must be located at the end of the statement*. This rule is included for the clarity of the requirement. An example of a requirement with the subject description clause located somewhere other than the end of the requirement statement is as follows.

> At GVW and GCW the vehicle shall pass the Jennerstown Brake tests.

In this requirement, the subject description clause is located at the beginning of the requirement sentence before the requirement subject. This phrase "at GVW and GCW" should be located at the end of the requirement as follows.

> The vehicle shall pass the Jennerstown Brake tests at GVW and GCW.

Using this rule, the main parts or ideas of the requirement (the subject, verb and object) are located at the beginning of the requirement in the same section without being interrupted by exceptions or description clauses.

Rule 6: *Requirements with clauses describing the direct object must be located immediately after the direct object*. This rule is included for clarity of the requirement. An example of a requirement with the direct object clause located somewhere other than after the direct object is as follows.

> If necessary to meet other requirements, a cab controlled tire pressure system shall be furnished, in accordance with Annex XX.

Though this is understandable, to again maintain a uniform requirement structure the requirement should be written as follows with the description clause at the end of the requirement statement.

> A cab controlled tire pressure system shall be furnished if necessary to meet other requirements, in accordance with Annex XX.

Rule 7: *Requirements with clauses that reference other requirements must be located at the end of the statement*. This is also a rule stated to give uniformity to the requirement layout. Upon study of approximately 160 FMTV requirements, it was found that references to other requirements were not stated in a specific place in the requirements but were rather scattered throughout. An example of an incorrectly stated requirement reference is as follows.

> In accordance with MIL-STD-XXXX, maintenance personnel shall not be exposed to concentrations of toxic gasses.

The correctly stated requirement is as follows moving the clause referencing other requirements to the end of the statement.

> Maintenance personnel shall not be exposed to concentrations of toxic gasses in accordance with MIL-STD-XXXX.

Rule 8: *Requirements should not be used to specify or describe a test.*

This is a rule designed to exclude extraneous information from the requirement text. Any information on the requirement test should be included in the test. Consider the following requirement.

> Test criteria cited in section 4 of this specification are to be considered minimum standards.

There is no physical system, subsystem or structure that can be used as the requirement. The subject and focus of this requirement is the test. This test should be excluded from the requirement list. Consider the next requirement example as well.

> The vehicle shall be tested and evaluated IAW section XX.

With this requirement, the vehicle is indeed the subject, but is still focused on the test. Since all requirements should have a test to ensure satisfaction [36], this requirement would have a test to ensure the vehicle was tested and evaluated according to IAW section 4.7.21.

Rule 9: *Requirements should be written in active, not passive voice.*

The difference between active and passive voice is whether the subject acts or is acted upon [37]. The subject of a passive voice verb is at the end of the sentence. This can be seen in the following example requirement.

Performance requirements shall be achieved with all models.

"Performance requirements" is the recipient of the action "achieved" from the object "models". To convert this requirement to an active voice, the subject and object must be switched and the verb tense changed from passive to active. Consider the following active voice requirement.

All models shall achieve performance requirements.

Notice the active voice is less wordy than the passive voice and the removal of the linking verb "be" changing the requirement from a nonfunctional requirement to a functional requirement.

Rule 10: *Requirements should always be written in the transitive or intransitive tense when possible*. This rule ensures that the verb 'be' is eliminated as much as possible in the requirement statement. Many times a functional requirement can masquerade as a non-functional requirement by using the verb 'be'. Functional requirements are those requirements that characterize the actions that the design must accomplish [11]. Non-functional requirements are requirements on the qualities of the design [38]. An example of a functional requirement that is worded like a non-functional requirement is as follows.

The rear view mirrors shall be *re-adjusting* without the use of tools.

The requirement is currently worded that re-adjusting is a quality attribute of the system. However, the re-adjustment of the mirrors is a function of the design, and is therefore a functional requirement. The requirement should be correctly worded as.

The rear view mirrors shall re-adjust without the use of tools.

The verb 're-adjust' is now a function of the design. By removing the word 'be' a correct functional requirement is produced.

Rules 1,2 and 3 were created in response to the way the Army represents requirements. To date, the granularity of Army requirements are entire paragraphs and no sentence analysis is done at all. For instance, consider military standard MIL-STD-961E(1), the US Military standard for preparing other military standards. This standard addresses how requirements should be written. In section 4.2 of this standard, it states that: "A specification shall be prepared to describe essential technical requirements for products, materials, or services. Similar items shall be covered in a single specification to the maximum extent practical. Specifications shall describe the item in a manner that encourages maximum competition. To the greatest extent possible, specification requirements shall be written so that commercial products or processes may be used to meet the requirements. Performance specifications shall be developed instead of detail specifications, whenever possible" [39]. This treatment of the requirements as an entire string leads to some inadvertent actions by the designer. First, because the requirement is treated like a text string, all burden of understanding the requirement, understanding its meaning and the relationships among design domains implied by this is laid upon the designer. The designer must look at the requirement sentence and somehow extract all (and no more) information that was put into it. This stems from the way a requirement is first written. Since there is no standard for stating a requirement, requirements are stated differently when stated by different people.

The US Army uses the MIL-STD-961E(1) to create the design requirements for the Family of Medium Tactical Vehicles (FMTV). This requirements document is known as the ATPD2131F.1. (In this document, requirements are called specifications.) While following the guidelines established in the MIL-STD document of a requirement describing “a single specification to the maximum extent practical”, this “single specification” could be a sentence or paragraph mentioning several parts of the system architecture along with several functions of those system parts. While being informative, this approach to stating requirements is somewhat limiting. Consider the following example requirement:

Table 4: Example Requirements to Establish the Fording Capabilities of FMTV vehicles taken from the ATPD2131F.1 requirement document

Fording. The vehicle shall be capable of operating in fresh and salt water in depths to XX without preparation. Fording for XX minutes shall not cause engine stall, damage or degradation of vehicle components, need for maintenance actions nor render the vehicle incapable of performing any operation of this specification. Excepted from this requirement are any non-sealed brake components. While fording, the engine shall be capable of being restarted when stopped for XX minutes. Seals shall restrict the entrance of foreign matter into bearings which are exposed to contamination during these operations. Water contamination of bearing lubricants shall not be more than XX by volume. All bearing seals shall restrict the leaking of lubricants from the bearings. Water contamination of engine, brake fluid, transmission, transfer transmission, power steering pump, fuel tank(s) and all differentials shall not exceed XX by volume. Vented components shall be vented above the XX-inch fording line without kit.

The fording requirement in Table 1 captures several different domains including:

- components and assemblies in the systems,
- functions in the system,
- qualifications and exceptions

Further, the requirement is complex because it implicitly captures and models the inter-relationships between these domains. In addition to the complexity of the requirement based on the number of functions and components that are constraints, the verbiage in the requirement is also difficult to fully comprehend – leading to increased complicatedness. The fording requirement constrains the following systems within the vehicle:

- entire vehicle
- engine
- vehicle components
- seals
- bearing seals
- brake fluid
- transmission
- power steering pump
- fuel tank
- differentials
- vented components

Additionally the requirement also captures several different functions, denoted by verbs, of the vehicle systems:

- capable of operating
- fording
- not stall
- not damage
- not degrade
- capable of being restarted.

In summary, there are eleven physical systems, six functions and one exception clause. In the physical world, these entities interact with each other and the study of their interaction is beneficial in terms of complexity studies and functional design. However, comparison of entities discussed in this requirement will be difficult since this requirement is a combination of all of these domains. Consider the following example problem in Figure 8:

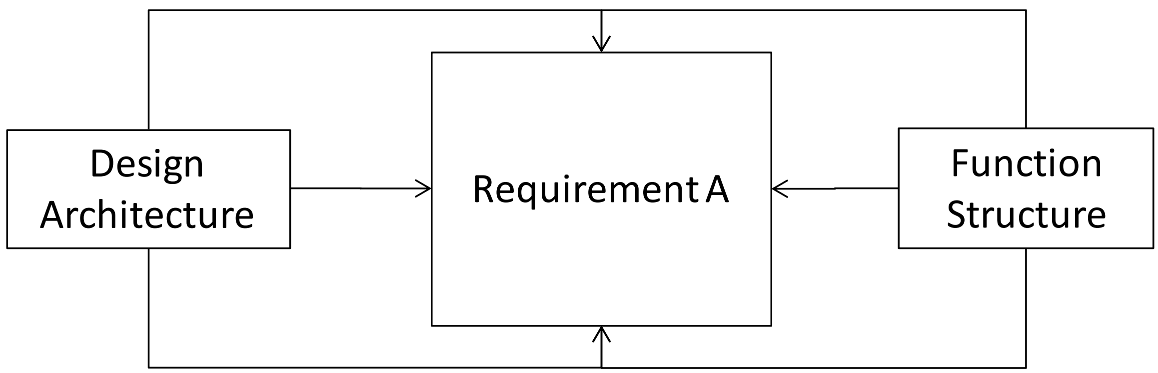


Figure 8: Mapping of Multiple Design Architectures and Functions to a Single Requirement

In Requirement A, multiple system/subsystem/components map to a single requirement. A similar situation exists with the mappings from the function structure to Requirement, meaning that there are multiple verbs or actions in the requirement. This presents a problem. It will be difficult to model the relationships between the several domains addressed in Requirement A since something on the order of an "Internal Requirement Diagram" will have to be constructed. Compare this situation with Figure 9.

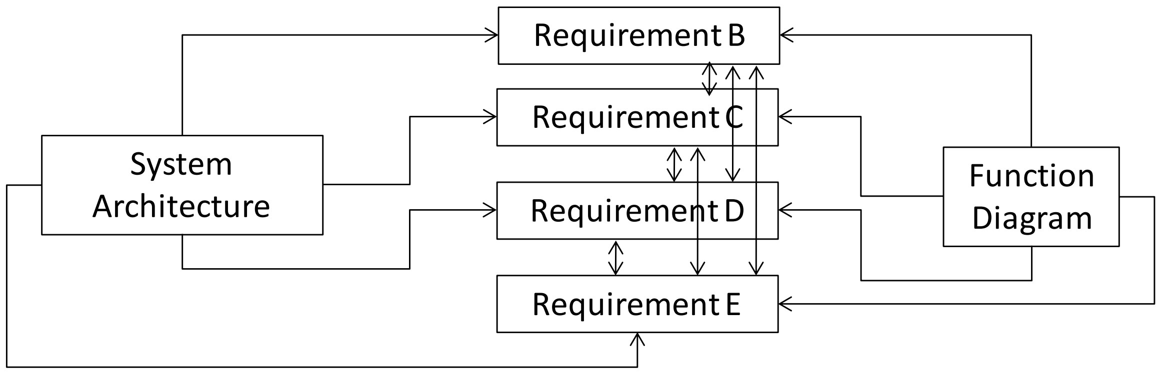


Figure 9: Mapping of Single Design Architectures and Single Functions to Single Requirements

Notice in this figure that each requirement has only one mapping from system architectures to applicable functions. Also, by decomposing the requirements, it becomes possible to map relationships between requirements, a necessary task in requirement mass reduction method described in the next chapter. Requirement rules 2 and 3 are used to split requirements into sentences that contain single subjects (system architectures) and verbs (functions).

By applying the preprocessing rules 2 and 3 discussed in Figure 9 to Requirement 3.2.1.7, the following requirements are created:

1. The vehicle shall be capable of operating in fresh water in depths to XX inch without preparation.
2. The vehicle shall be capable of operating in salt water in depths to XX inch without preparation.
3. Fording for XX minutes shall not cause engine stall, except for any non-sealed brake components.
4. Fording for XX minutes shall not cause engine damage except for any non-sealed brake components.

5. Fording for XX minutes shall not cause degradation of vehicle components except for any non-sealed brake components.
6. Fording for XX minutes shall not cause engine maintenance actions except for any non-sealed brake components.
7. Fording for XX minutes shall not render the vehicle incapable of performing any operation of this specification except for any non-sealed brake components.

The rules stated in 4, 5, 6 and 7 are stated to consistently place clauses in specific places. If these rules were not stated, then subject description clauses would exist at the beginning of the requirement in some requirements and at the end of the requirement statement in other requirements. It could be just as valid if these rules state that the several clauses be placed at the beginning of the requirement statement. By consistently stating the requirement clauses, a limited level of automation can be included in the requirement design process. The pre-processing rules have explained <u>what</u> exactly goes into the requirement. It must be explained <u>how</u> the requirement is to be represented syntactically.

Syntactical Rules

The requirement syntactical rules developed by Lamar will be used to dictate how the requirement information is displayed in the requirement sentence [32]. These syntactic rules are language-based, meaning that different languages will have different syntactical rules based on types and orders of subjects, verbs and adjective phrases. Since syntactic rules are the mechanism used to formulate requirements, they are vital to mass reductions analysis techniques.

The parts of speech used in English are the subject, modal verb, main verb and the adjective (or adjunct) phrase. According to Lamar [32], the syntactic parts of speech definitions are included in Table 5 and the sentence structures of a requirement are shown and discussed in the following pages.

Table 5: English parts of speech

Part of Speech	**Definition**
Subject	refers to the part of the system that must comply to a specific parameter
Modal Verb	verb that refers to “shall, should, must”
Main Verb	describes what the subject must do in case of action verbs or links the quality the subject must have in the case of linking verbs
Adjunct Phrase	describes another word or phrase

Table 6: Transitive functional requirement sentence structure

<functional requirement> = <subject> “modal” <main> {<adjunct>}

Table 7: Intransitive functional requirement sentence structure

<functional requirement> = <subject> “modal” <main verb> {<direct object>} {<adjunct>}

Table 8: Nonfunctional requirement sentence structure

<nonfunctional requirement> = <subject> “modal” <linking verb> <subject complement>} {<adjunct>}

Table 6, Table 7 and Table 8 show the general layout of different types of requirement sentences. The subject of the sentence pertains to the person or thing the sentence is about [40]. This is illustrated in Table 9.

Table 9: Subject of a requirement sentence

The <u>radiator</u>	shall	minimize	air side fouling	by location.
Subject	*modal verb*	*main verb*	*object*	*adjective phrase*

In this sentence, the subject of the sentence is “radiator”, meaning that the thing the sentence (or in this case the engineering requirement) is about is the radiator component.

There are two types of verbs that follow the subject: modal verbs and main verbs. The modal verb shows the level to which the requirement shall be met [32]. The main verb can be one of two different kinds: an action verb or a linking verb. Action verbs describe what the subject is doing and linking verbs describe the subject by linking the subject with the object of the sentence [40].

According to Berk [40], there are two types of action verbs, transitive and intransitive verbs. *Transitive* verbs require an object of the sentence to complete the predicate. Objects are the noun phrases that follow the verb. This is illustrated in Table 10.

Table 10: Transitive verb of a requirement sentence

The MOS-designated drivers	shall	<u>control</u>	the vehicle.
subject	*modal verb*	*transitive verb*	*object*

In this sentence, the word “control” is the transitive verb. The sentence would not make sense if it read “The MOS-designated drivers shall control.” The sentence needs a noun phrase (or object) for completion.

Intransitive verbs, unlike transitive verbs, do not require an object to complete the sentence. Direct objects may still be used, but are not needed to make the sentence complete. An example of an intransitive functional requirement is shown in Table 11.

Table 11: Intransitive verb of a requirement sentence

The transmission	shall	<u>shift</u>.
subject	*modal*	*intransitive verb*

"Shift" is the intransitive verb and does not require an object to complete the sentence. An example of an intransitive requirement sentence that has an adjunct is shown in Table 12.

Table 12: Intransitive verb of a requirement sentence with adjunct

The transmission	shall	<u>shift</u>	in the forward and reverse gear.
subject	*modal*	*intransitive verb*	*adjunct*

While the adjunct adds more detail and information to the requirement sentence, it is not needed to make the sentence grammatically correct. An example of a linking verb is included in Table 13.

Table 13: Linking verb of a requirement sentence

The oil sampling valves	shall	<u>be</u>	usable	while the engine is running.
Subject	*modal verb*	*linking verb*	*object*	*adjunct*

Engineering requirements are grouped into two types: functional and non-functional requirements. Functional requirements are requirements dictating the actions of the system. Nonfunctional requirements are requirements on the system attributes [38]. Functional requirements can include either a transitive or intransitive verb and they are represented as follows.

Table 14: Functional requirement sentence with an intransitive verb

The vehicle systems shall start in the ambient temperature range of 120 °F to -25 °F.

In Table 14, the function of the requirement is "start". This is also intransitive since it does not require an object for completion. The sentence would be complete if read "The vehicle systems shall start". A transitive functional requirement is illustrated in Table 15.

Table 15: Functional requirement sentence with a transitive verb

The cooling system	shall	recover	xxxx % coolant overflow.
subject	*modal*	*verb*	*adjective phrase*

"Recover" is a transitive verb that must have an object to clarify what is meant: thus the object "xxxx% coolant overflow."

Nonfunctional requirements are requirements on the attributes of a system and therefore have a linking verb. They can be written in the following manner.

Table 16: Nonfunctional requirement sentence with a linking verb

Components shall be protected from corrosion by scheduled maintenance.

In Table 16, the attribute describing the subject is "protected" and is linked to the subject by the verb "be".

These pre-processing rules and syntactical rules are applied to approximately 160 FMTV requirements and converted to approximately 800 consistently stated FMTV requirements. A snippet of the requirement analysis is shown in Table 17.

The "FMTV Heading" corresponds to the number of the requirement in the ATPD2131F document. In the ATPD, each requirement corresponds to a paragraph. After processing, multiple requirements were decomposed from a single ATPD requirement number. This was the case for requirement 3.2.1.5 in Table 17. Even though the second requirement stated does not have a corresponding number beside it, it still belongs to requirement 3.2.1.5 in the ATPD. The "Requirement (original)" corresponds to the raw unprocessed requirement from the ATPD. The "Requirement (processed)" corresponds to the requirement after being processed using the pre-processing and syntax rules. The "Subject", "Verb", "Object" and "Adjective Phrase" columns correspond to the parts of speech each word or phrase corresponds to. The "Verb Type" column refers to the type of verb (transitive or intransitive) that each requirement verb belongs to. The "Requirement Type" column corresponds to the type of requirement it is. Requirements with action verbs are Functional Requirements. Requirements with linking verbs are Nonfunctional Requirements. This process was accomplished for all 800 FMTV requirements and can be viewed in Appendix 2.

FMTV Heading	Original Requirement	Processed Requirement	Subject	Verb	Object	Adjective Phrase	Verb Type	Requirement Type
3.2.1.5	The Wall to Wall turning radius for the LMTV shall not exceed XX feet in one continuous movement.	The FMTV shall have a turning radius less than XX feet in one continuous movement.	FMTV	Have	turning radius	less than XX feet in one continuous movement	Transitive	Non Functional
	Vehicle with stated towed items shall be capable of turning, in one continuous movement, two standard NATO intersection or XX walled intersection roads at XX°.	The vehicle shall turn, two standard NATO XX intersection or XX walled intersection roads at 90°, in one continuous movement, upon transporting stated towed items.	vehicle	turn	NATO XX intersection road, XX walled intersection road	in one continuous movement, upon transporting stated towed items	Intransitive	Functional
3.2.1.11	All vehicles (trucks and trailers) shall be equipped with an Anti-lock Brake System (ABS) and shall comply with FMVSS XX.	All vehicles (trucks and trailers) shall be equipped with an Anti-lock Brake System (ABS).	vehicles	be	equipped	with an Anti-lock Brake System (ABS)	Linking	Non Functional
		All vehicles (trucks and trailers) shall comply with FMVSS XX.	vehicles (trucks and trailers)	comply	FMVSS XX		Transitive	Non Functional
	Brake linings shall be constructed from non-asbestos materials.	Brake linings shall be constructed from non-asbestos materials.	Brake Linings	be	constructed	non-asbestos materials	Transitive	Non Functional

Table 17: Screenshot of FMTV requirement analysis

Step 2: Map requirements to components

Create RxC and RxR Matrices

Step 2 involves creating relationships between the requirement and component domains. Relationships between entities in a design domain can be displayed and analyzed in matrix form using a Design Structure Matrix (DSM). Design domains are any single aspect of a design. For instance, the design requirements are a design domain. The design architecture (system, subsystem, components) would be another design domain. Elements within a domain could be specific requirements or components within a design. DSM matrices have identical rows and columns that show the couplings between entities inside a single domain as shown in Table 18 [41].

Table 18: DSM matrix of elements in the same domain

		LETTERS					
		A	B	C	D	E	F
LETTERS	A						
	B	x		x	x		x
	C	x	x			x	x
	D	x	x			x	x
	E	x		x	x		
	F	x	x	x	x	x	

In Table 18, the intersection of a row element and a column element is the possibility of a relationship between the two elements. An "x" denotes a relationship between two entities. If no "x" exists here, then there is no relationship between the two elements. Since each element is related to itself, the diagonal consists of all "x's". For example, element A is completely decoupled since it is not related to other elements, element C is coupled since it is related to other elements and element F is completely coupled since it is related to every other element. DSMs can be used to identify uncoupled elements that, if changed, would not affect other elements in that particular domain.

Different design domains can also be related to each other. Matrices that illustrate this relatedness are called Domain Mapping Matrices (DMM). These matrices are usually rectangular since the number of elements in each domain is often not the same [42]. Table 19 shows an example DMM matrix between two domains: NUMBERS and LETTERS.

Table 19: DMM matrix of elements of two domains

	LETTERS					
NUMBERS		A	B	C	D	E
	1	x			x	
	2		x	x	x	
	3	x				
	4	x	x	x	x	x
	5					

Notice that there is no identity matrix in a DMM since the elements related are not from within the same domain. Element 1 is related to elements A and D in this matrix. Element 4 is completely coupled to all entities in the LETTERS domain. Element 5 is completely uncoupled from all elements in the LETTERS domain.

The strength of the relationship between two elements can vary. Sometimes if a relation either exists or does not, a binary relation is needed. This is represented using either 1's or 0's. Other times, relationships can have different strengths. This can be represented by using numbers other than 1's or 0's like 0's (no relationship), 1's (weak relationship), 3's (medium relationship) and 9's (strong relationship). Any range or granularity of strengths can be used, the important part is the difference between the ranking numbers.

Requirements can be related to components through a number of ways: through subjects, verbs or adjective phrases. Due to the structured way that the requirements are written using the pre-processing rules, there is a direct mapping between requirements to components through the subject. Namely, the requirement is related to the component mentioned in the subject and can be codified in two Requirement/Component Relationship Rules:

1. The requirement is related to the component mentioned in the subject. If the subject of the requirement is not a leaf node of the design structure hierarchy (a branch), then it affects all components that are lower than that point of the physical hierarchy.
2. If a requirement refers to "all components" or to "all materials" or to properties of components, the requirement is linked to all the leaf nodes of the physical hierarchy.

For example, if a system was modeled like the one shown in Figure 10, then a requirement that states, "Component 1 shall be red" is related to and only to component 1 because that is the only part of the design mentioned in the requirement.

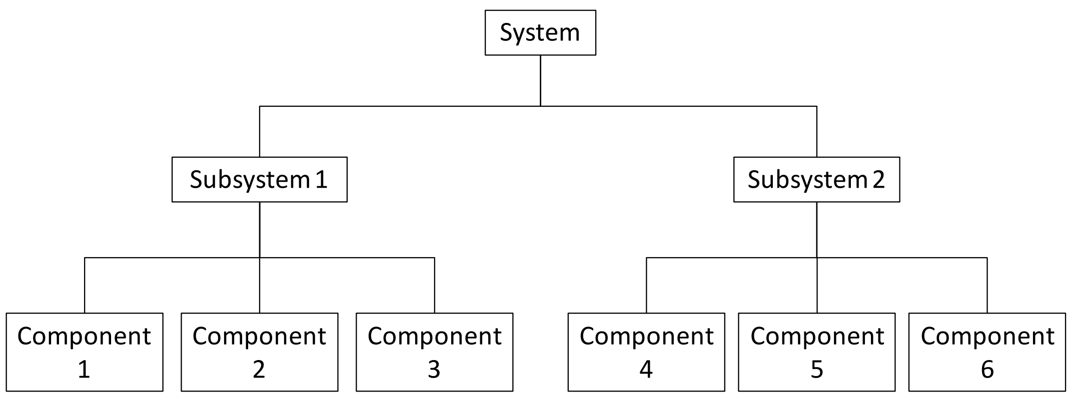


Figure 10: Example design illustration for inter-domain relationships

If another requirement read, "Subsystem 1 shall be recyclable", then the requirement pertains to subsystem 1 and also all the other subsystem/components that branch off of subsystem 1, in this case, component 1,2 and 3. If another requirement read "All components shall be safe", the requirement would pertain to components 1,2,3,4,5 and 6. An example design system will be used to illustrate the method discussed in this chapter. While FMTV requirements were used as examples for the preprocessing rules, a subsystem was not available to use as an example of steps 2,3 and 4 of the analysis method. Thus, components and component masses used here are from a BMW engine liquid cooling subsystem.

In this step, the component parts and their corresponding masses are also attained. This can be accomplished in several ways. The most straightforward way would be to physically weigh the components. If physical parts are not available and part files are, another way to measure the mass of components is to use CAD tools. This is the approach used in this report. Once this is accomplished, requirements are then related to components.

In step 2, the two types of matrices are constructed: one using 1's and 0's and the other using the component's mass as a relational strength. The transpose of the Requirement vs. Component matrices are taken (creating Component vs. Requirement matrices) and Requirement vs. Requirement matrices constructed. The Requirement vs. Requirement matrices are constructed by multiplying the RxC mass matrix with the transpose of the RxC binary matrix. This is shown in Equation 3.1.

$$RxC\,Mass \times CxR\,Binary = RxR\,Mass \qquad 3.1$$

The requirements and components are mapped using a DMM. The DMM of the BMW cooling subsystem using 1's and 0's is shown in Table 21. Using the mass list in Table 20, the mass list is incorporated into Table 22 to create the RxC mass matrix. The DMM using mass as the relationship is shown in Table 22.

Table 20: BMW cooling subsystem component mass list

BMW Cooling System Component	Mass
Fan	0.7
Thermostat	0.083
Expansion Tank Subassembly	0.884
Radiator Cap	0.04
Radiator Subassembly	5.0
Inlet Water Hose	0.3
Outlet Water Hose	0.3
Temperature Sensor	0.02
Water Pump Subassembly	4.68
Engine Coolant	4.92
Oil Cooler	0.6
Drying Container	0.3
Condenser Subassembly	2.2

Table 21 : Requirement vs. Component matrix using 1's and 0's

	Fan	Thermostat	Expansion Tank Subassembly	Radiator Cap	Radiator Subassembly	Inlet Water Hose	Outlet Water Hose	Temperature Sensor	Water Pump Subassembly	Engine Coolant	Oil Cooler	Drying Container	Condenser Subassembly
R1: Hoses shall have quick fit connectors.	0	0	0	0	0	1	1	0	0	0	0	0	0
R2: Hoses shall be mix-up proof.	0	0	0	0	0	1	1	0	0	0	0	0	0
R3: Coolant shall have a temperature between -40°C to +140°C .	0	0	0	0	0	0	0	0	0	1	0	0	0
R4: The cooling system shall have pressures between 18mbara to 3.5bara.	1	1	1	1	1	1	1	1	1	1	1	1	1
R5: The cooling system should use common parts internally and externally.	1	1	1	1	1	1	1	1	1	1	1	1	1
R6: The radiator mesh shall have a total frontal area of mesh of 580mm x 449mm.	0	0	0	0	1	0	0	0	0	0	0	0	0
R7: The oil cooler shall have dimensions of block size X=45mm; Y=165mm; Z=80mm.	0	0	0	0	0	0	0	0	0	0	1	0	0
R8: The engine coolant pressures shall be in the range of 3bara at –40°C to +143°C.	0	0	0	0	0	0	0	0	0	1	0	0	0
R9: The mesh condenser shall have a total frontal area of 22.2dm2.	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 22: Requirements vs. Components matrix using component mass as the relational strength

	Fan	Thermostat	Expansion Tank Subassembly	Radiator Cap	Radiator Subassembly	Inlet Water Hose	Outlet Water Hose	Temperature Sensor	Water Pump Subassembly	Engine Coolant	Oil Cooler	Drying Container	Condenser Subassembly
Mass	0.7	0.083	0.884	0.04	5	0.3	0.3	0.02	4.68	4.92	0.6	0.3	2.2
R1	0.00	0.00	0.00	0.00	0.00	0.30	0.30	0.00	0.00	0.00	0.00	0.00	0.00
R2	0.00	0.00	0.00	0.00	0.00	0.30	0.30	0.00	0.00	0.00	0.00	0.00	0.00
R3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.92	0.00	0.00	0.00
R4	0.70	0.08	0.88	0.04	5.00	0.30	0.30	0.02	4.68	4.92	0.60	0.30	2.20
R5	0.70	0.08	0.88	0.04	5.00	0.30	0.30	0.02	4.68	4.92	0.60	0.30	2.20
R6	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
R7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.00	0.00
R8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.92	0.00	0.00	0.00
R9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.20

The mapping relationships in Table 21, indicated by 1's and 0's represent requirements that have an influence on a particular component. This mapping shows the existence of a relationship and not the relationship's strength. For example, referring back to Table 21, the requirement

R4: The cooling system shall have pressures between 18mbara to 3.5bara.

is equally mapped to thirteen components. In reality, the requirement may have a greater influence on a subset of the components. To identify weighted relationships, the component mass is used as the relationship strength.

The component mass information is combined with a Requirements vs. Components matrix in Table 22. This more accurately gives a description of which requirements affect the most total mass. For example, referring back to Table 22, the requirement

R4: The cooling system shall have pressures between 18mbara to 3.5bara.

is mapped to thirteen components with significant weighting differences. The mass identifies the maximum amount of mass that the requirement can affect in that component. In reality, the requirement may not affect all of a component's mass.

From the requirements to components binary mapping matrix, it is possible to identify the highly influential requirements based on the number of components that each requirement affects. This is simply the sum of each row. This is done in step 3 of this process.

The relationships between requirements for the cooling systems are determined based on the requirement to component mapping matrix. The Requirement vs. Requirement matrix is computed by multiplying the RxC matrix by its transpose.

For this thesis, multiplying the RxC mass matrix by the CxR binary matrix will be used. This is shown as follows. The CxR binary matrix is shown in Table 23.

Table 23: CxR binary matrix for the BMW cooling subsystem

	R1	R2	R3	R4	R5	R6	R7	R8	R9
Fan	0	0	0	1	1	0	0	0	0
Thermostat	0	0	0	1	1	0	0	0	0
Expansion Tank Subassembly	0	0	0	1	1	0	0	0	0
Radiator Cap	0	0	0	1	1	0	0	0	0
Radiator Subassembly	0	0	0	1	1	1	0	0	0
Inlet Water Hose	1	1	0	1	1	0	0	0	0
Outlet Water Hose	1	1	0	1	1	0	0	0	0
Temperature Sensor	0	0	0	1	1	0	0	0	0
Water Pump Subassembly	0	0	0	1	1	0	0	0	0
Engine Coolant	0	0	1	1	1	0	0	1	0
Oil Cooler	0	0	0	1	1	0	1	0	0
Drying Container	0	0	0	1	1	0	0	0	0
Condenser Subassembly	0	0	0	1	1	0	0	0	1

This table is the transpose of the RxC binary matrix. When the RxC mass matrix is multiplied by the CxR binary matrix, the following RxR mass matrix is obtained. The resulting RxR matrix is shown in Table 24.

Table 24: RxR mass matrix for BMW cooling subsystem

	R1	R2	R3	R4	R5	R6	R7	R8	R9
R1	0.60	0.60	0.00	0.60	0.60	0.00	0.00	0.00	0.00
R2	0.60	0.60	0.00	0.60	0.60	0.00	0.00	0.00	0.00
R3	0.00	0.00	4.92	4.92	4.92	0.00	0.00	4.92	0.00
R4	0.60	0.60	4.92	20.03	20.03	5.00	0.60	4.92	2.20
R5	0.60	0.60	4.92	20.03	20.03	5.00	0.60	4.92	2.20
R6	0.00	0.00	0.00	5.00	5.00	5.00	0.00	0.00	0.00
R7	0.00	0.00	0.00	0.60	0.60	0.00	0.60	0.00	0.00
R8	0.00	0.00	4.92	4.92	4.92	0.00	0.00	4.92	0.00
R9	0.00	0.00	0.00	2.20	2.20	0.00	0.00	0.00	2.20

Table 24 shows the results for relating requirements to each other by mass. This process is initiated using rules for relating requirements to components and automated through matrix multiplication. This process minimizes human error due to populating requirement matrices.

Step 3: Requirement Analysis and Identification of Mass Intensive Requirements

The requirement analysis in this section focuses on two primary types of coupling:

1. Coupling to requirements (number of requirements coupled to)
2. Coupling to mass
    a. Coupling to mass by one requirement
    b. Coupling to mass through other requirements

This is obtained by using the two different types of matrix relationships, binary and mass. To find information related to how much mass a requirement affects, refer to the mass matrices. To find information related to how many requirements a requirement affects, refer to the binary matrices.

Step 3 will explain how to obtain the desired requirement information from each of these matrices.

To identify requirement coupling, the RxR binary matrix must be examined. This was accomplished in Step 2 but was not shown. For illustration purposes, it is shown here.

Table 25: RxR binary matrix for BMW cooling subsystem

	R1	R2	R3	R4	R5	R6	R7	R8	R9
R1	2	2	0	2	2	0	0	0	0
R2	2	2	0	2	2	0	0	0	0
R3	0	0	1	1	1	0	0	1	0
R4	2	2	1	13	13	1	1	1	1
R5	2	2	1	13	13	1	1	1	1
R6	0	0	0	1	1	1	0	0	0
R7	0	0	0	1	1	0	1	0	0
R8	0	0	1	1	1	0	0	1	0
R9	0	0	0	1	1	0	0	0	1

The diagonal values (in light gray) in Table 25 show the number of components affected by each requirement. For example, two components are affected by R2. The non-diagonal values show the number of components affected by two requirements. For example, two components are affected by R2 and R4 (shown in dark gray). To find the number of requirement couplings using the RxR binary matrix, two requirements are coupled if they affect at least one component, excluding the matrix diagonal. For instance, R8 affects three components.

Each cell in Table 24 shows the maximum amount of mass affected by those two requirements. The diagonal (light grey cells) shows the most mass affected by each requirement alone. So, the total mass affected by R3 is 4.92 kg. All off-diagonal cells (white cells other than labels) show the mass affected by the combination of two requirements. For example, the mass affected by R4 and R9 is 2.20 kg. of mass. The mass affected by one requirement and all other requirements is calculated by summing across a row. For example, the total mass affected by R6 and all other requirements is 15.0 kg. of mass.

Using these matrices, the following information in Table 26 can be extracted from the BMW cooling system requirements.

Table 26: BMW requirement analysis for requirement coupling and mass coupling

	total mass affected by 1 requirement (kg)	total mass coupled to (w/other requirements) (kg)	# requirements related to
R1	0.60	1.80	3
R2	0.60	1.80	3
R3	4.92	14.76	3
R4	20.03	38.87	8
R5	20.03	38.87	8
R6	5.00	10.00	2
R7	0.60	1.20	2
R8	4.92	14.76	3
R9	2.20	4.40	2

It is important to note the advantages and disadvantages of the resulting matrix. The subsequent sections will use the RxC and RxR matrices created in this section.

Identify Uncoupled Requirements

When manipulating requirements, requirements with high couplings to other requirements present a problem since making a change to one requirement also makes a change to the other requirements it affects. While sometimes beneficial, many times it is problematic, causing the designer to adjust other requirements so that they are all compliant. Thus, minimizing requirement coupling would be ideal since a change in one requirement would not affect other requirements (at best) or only a few other requirements (at worst). From the analysis in Step 3, shown in the fourth column in Table 26, no requirements are completely uncoupled from each other but three requirements are coupled to only two other requirements. Since these are the lowest couplings, requirements R6, R7 and R9 are the most uncoupled requirements.

Identify Requirements Coupled to Mass

To find requirements that affect the most mass, the RxR mass matrix is used and the diagonal values are reviewed. From the second column in Table 26, it is shown that R4 and R5 affect significantly more mass than others at 20.03 kg. Thus R4 and R5 are the requirements that singly affect the most mass. To find requirements that, coupled to other requirements together affect the most mass, the rows (excluding the diagonal) are added together. The results are shown in the third column in Table 26. As far as determining the size of the "acceptable" set of requirements to change, this is for the user to decide. In this thesis, the set size was made at definite breaks in the data. For instance, only R4 and R5 were chosen because the other requirement weights were significantly lower, the next one starting at 5.00 kg.

Taking into account mass and coupling leads to several different combination possibilities: requirements that affect much mass and are also highly coupled, requirements that affect much mass and are lowly coupled, requirements that affect little mass and are highly coupled and requirements that affect little mass and are lowly coupled. Also to be included are mid-level mass or coupling values. These are always treated as second-choice options if the best case option is not available. This is graphically shown in Figure 11.

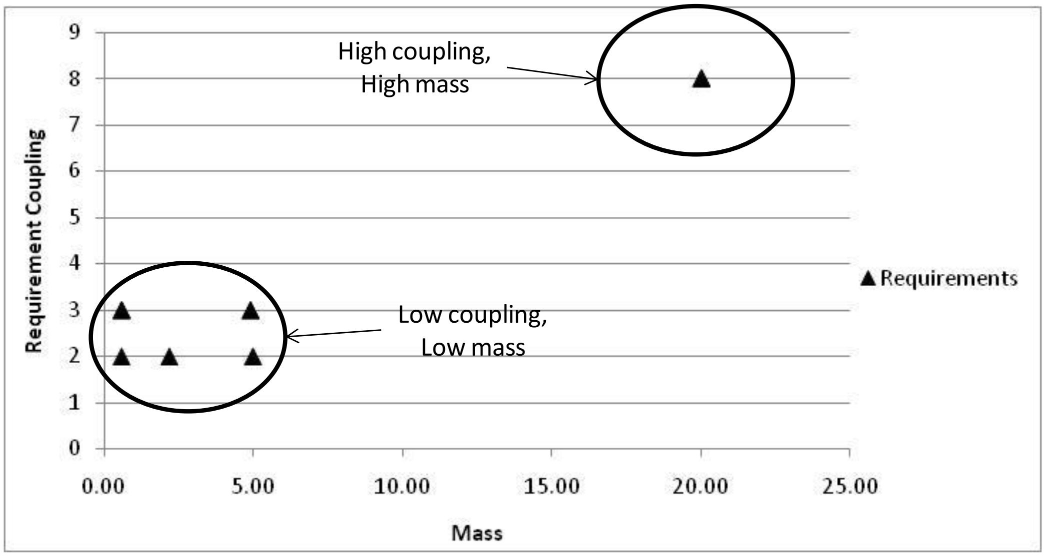


Figure 11: Requirement coupling vs. mass for BMW cooling subsystem

This figure shows the cooling requirements fit into two categories: high mass, low coupling and low coupling, low mass. Each data point stands for the group of requirements that have the same values for requirement mass and coupling. For example, the high coupling high mass data point represents R4 and R5 from the requirements list which both have mass values of 20.03 and coupling values of 8.

Selecting the requirements to change comes by examining the categories the requirements can be grouped in. The most desirable would be to have requirements that affect much mass and are also lowly coupled. Suh also mentions the desirability of low coupled requirements in his Axiom #1: maintain the independence of requirements [11]. These are requirements that can be changed without affecting other requirements. The second most desirable group would be requirements that affect high mass and are also highly coupled. Reducing mass by changing these requirements could come at a cost, however. Because these requirements are highly coupled, changing these could cause other inadvertent changes in other requirements. These requirements can still be changed, but they are more labor-intensive to change. The next group of requirements to change would be requirements that affect little mass and are lowly coupled. These requirements may be easy to change but they affect little overall mass. The last group and most undesirable to change would be requirements that affect little mass and are also highly coupled. These requirements may give more trouble and inconvenience through their high coupling to other requirements than their benefit from reducing mass.

One field of requirements that these groups do not include are requirements that affect a "middle" level of mass and have a "middle" coupling level. These can be described as being better requirements to change than the "low" ones but are less of a priority to change than "high" requirements.

Given this discussion of how requirements are selected to change, the following Table 27 shows the order in which requirements should be modified to most efficiently reduce mass.

Table 27: Order priority for which requirements to change of BMW cooling subsystem

	total mass affected by 1 requirement (kg)	# requirements related to
R4	20.03	8
R5	20.03	8
R6	5.00	2
R3	4.92	3
R8	4.92	3
R9	2.20	2
R7	0.60	2
R1	0.60	3
R2	0.60	3

Requirements that affect high mass and are highly coupled are changed first while requirements that affect little mass and are lowly coupled are changed last.

Chapter Summary

This chapter outlined the proposed requirements analysis method in 3 steps,

1. Acquire and process requirements

2. Map requirements to components
3. Identify Requirements that are uncoupled and affect significant amounts of mass

Step 1 acquires the raw requirement and ensures they are stated according to the 10 pre-processing rules and syntax format. Step 2 maps requirements to components and generates the requirements to requirements matrix. Step 3 analyzes requirements and identifies mass intensive requirements that are also uncoupled. This process is shown again in Figure 12.

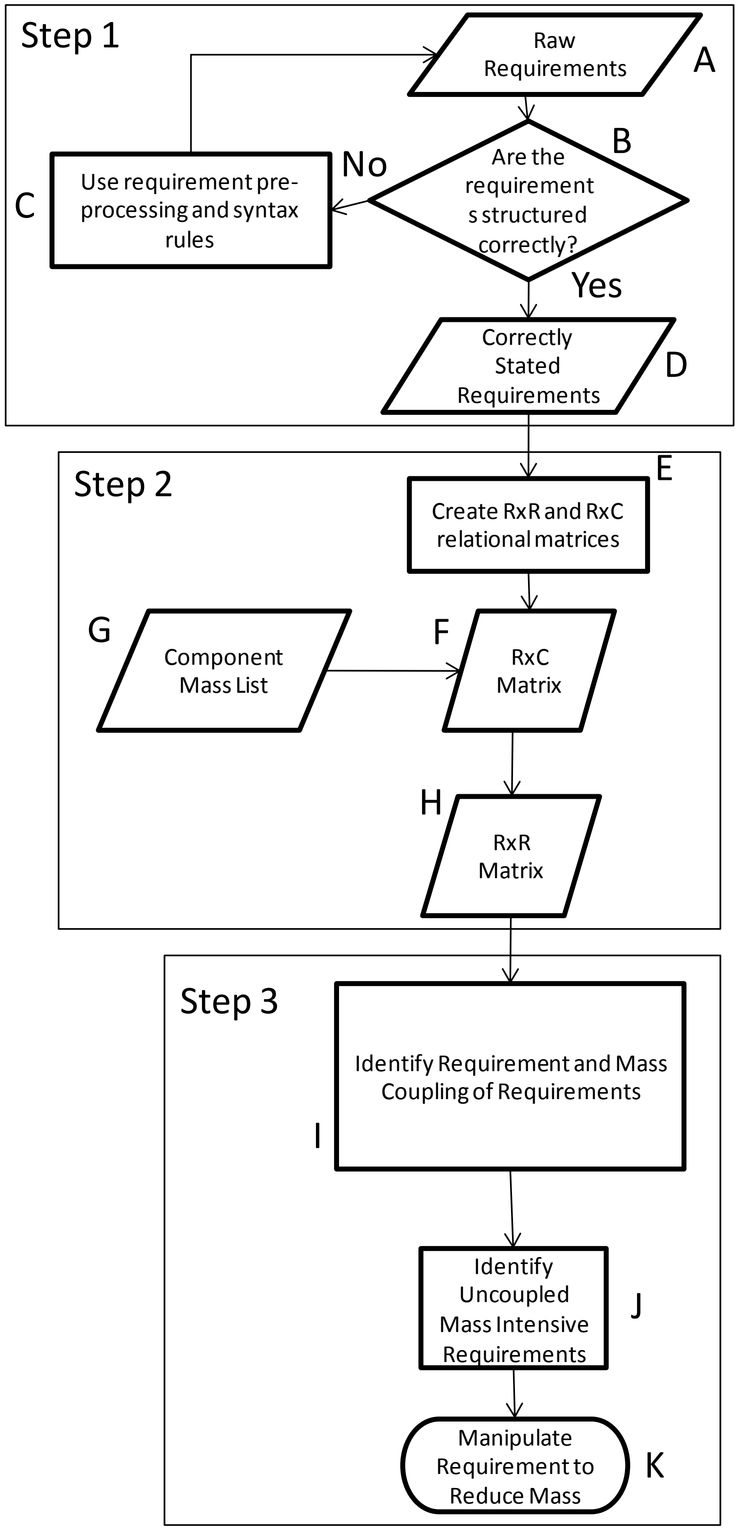


Figure 12: Flowchart of requirement analysis method

In the next chapter, several software applications are evaluated. This will be done by developing a metric to evaluate requirements software based on the inherent qualities of engineering requirements.

Chapter 4. Computer Implementation-Representing Requirements in a Computer-Based Environment

Several software tools have been created to model requirements. In this chapter, a software tool will be selected to implement the method discussed in Chapter 3. Two requirement engineering softwares are compared to the "capabilities" of engineering requirements identified in Chapter 2. This chapter will provide an explanation for using requirements design software tools to implement the requirement analysis method.

The development of complex vehicle systems spans several designers, times and locations. To successfully support the design of such systems, information technology is used to manage, share, and control design information across the extended product development team. Specifically, requirements management software has been of significant interest with larger projects where requirement storage, management and availability to many people is crucial to the success of the project. These software tools can be used as a means of arranging and storing design requirements. There are several different commercially available and research-based requirements management and modeling tools. The International Council on Systems Engineering (INCOSE) conducted a survey of multiple requirements management tools in regards to various requirement qualities. A snippet of this report is shown in Figure 13.

Tools:	CASE Spec 8.0	CARE 3.2	Compuware Optimal Trace	CORE 5.1	Cradle 5.2	Envision VIP	Gatherspace	IBM Rational RequisitePro (updated 10 Oct 06)	IRQA 4	KollabNet Editor 2005
Response Date	June 12, 2008	July 26, 2005	February 08, 2007	July 07, 2006	May 18, 2005	March 01, 2005	July 25, 2005	June 16, 2004	June 24, 2008	July 25, 2005
1. Capturing Requirements/Identification	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
1.1. Input document enrichment/analysis	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
1.1.1 Input document change/comparison analysis	Full	Part	Full	Full	Full	Full	Part	Full	Full	Part
1.2 Automatic parsing of requirements	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
1.3 Interactive/semi-automatic requirement identification	Full	Part	Full	Full	Full	Full	Part	Full	Full	Full
1.4 Manual requirement identification	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
1.5 Batch-mode operation	Full	Part	Full	Full	Full	Full	Full	Full	Full	Full
1.6 Requirement classification	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
2. Capture System Element structure (if so, how? As document paragraphs? product structures?, ...)	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
2.1 Graphically capture systems structure	Full	Part	Full	Full	Full	Full	Full	Full	Full	Full
2.2 Textual capture of system structure	Full	Full	Full	Full	Full	Full	None	Full	Full	Part
3. Requirements Flowdown	Full	Full	Full	Full	Full	Full	Part	Full	Full	Full
3.1 Requirements derivation (req. to req., req. to analysis/text)	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full

Figure 13: Excerpt from INCOSE requirement tool survey

Information to complete the survey was provided for twenty-five tools by the vendors of each tool. This brings to light several issues included biased responses from each tool vendor and a lack of standardized test case for evaluating the capabilities of each tool. The survey provides a solid foundation on which to evaluate requirements engineering software, but does not represent the current landscape and technology changes. Notably, the survey does not fully address the development of the Systems Modeling Language (SysML) and changes in requirements modeling tools. Two tools are evaluated in this research because of their widespread use in industry and academic research. These tools are DOORS (IBM) and MagicDraw SysML (No Magic). The evaluation of MagicDraw includes an evaluation of the SysML modeling language. These tools are utilized because of the widespread use of requirement repositories like DOORS (used in automotive and aerospace industries) and tools like MagicDraw (used in architecture and software engineering) which use the SysML framework created by the Object Management Group, an international computer industry consortium [43]. Unlike the INCOSE survey, the tool evaluation presented in this chapter is based on a standard design problem that implements the requirement capabilities discussed in Chapter 2 and requirements identified in current research. The design problem is implemented in each software tool. From this initial evaluation, the chosen software models three subsystems of the FMTV vehicle in Chapter 6, Chapter 7 and Chapter 8. It is important to note that Microsoft Excel is used in conjunction with the specialized requirements management software used for matrix calculations. Microsoft Excel is not evaluated in this study because it is a general purpose spreadsheet software and does not offer specialized capability for modeling engineering requirements.

In general, requirements management software tools must support eight key characteristics. These characteristics were discussed in Chapter 2 and are again listed here in Table 28 for reference in this chapter.

Table 28: Requirement specifications

Capability	Definition	Ref.
Refinement	Create requirements of narrower scope and higher specificity from parent requirements.	[14]
Requirement Traceability	Description of a requirement's evolution through the design process.	[15,16]
Satisfaction	Relationship between a requirement and the artifact designed to fulfill the requirement.	[18,19]
Verification	Relationship between a requirement and the test that ensures the requirement has been satisfied.	[20]
Coupling	Interrelationships between requirements.	[16,19]
Prioritizing	Importance ranking of a requirement.	[22]
Input Validation	Ensures quality, structured requirements are input into the software.	[23]
View Restrictions	Restricts specific users to viewing a subset of the total requirements.	[16,24]

In the following section, the demonstration problem is presented.

Air Induction System Demonstration Example Problem

To demonstrate the capabilities of requirements management tools, a requirements document and design specifications were obtained for the U.S. Army's Family of Medium Tactical Vehicles (FMTV). The FMTV consists of fourteen different vehicle types based on a common platform (see Figure 14).

Figure 14: FMTV Models 1080 A1 (2.5-ton) and 1092 A1 (5.0- ton) (source:[17,18])

The Technical Data Package (TDP) document is the source for engineering requirements information [20]. The TDP contains a mix of system-level, component-level, and verification tests. The requirements define the physical and performance characteristics of the FMTV. The TDP provides several different types of information about the system including [21]:

- 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.

In addition to the TDP, geometric CAD models were obtained for the entire vehicle. The CAD models were used to obtain information about components and assemblies. The TDP contains approximately 150 complex requirements and several thousand geometric models. The baseline example is developed from design documentation for the FMTV. Specifically, the air induction subsystem is chosen as the system to model and analyze. The example includes five engineering requirements, one validation test, and four components. In addition, there are several inter-relationships between the requirements, tests, and components that are modeled. A detailed description is provided in the following sections.

FMTV Air Induction Systems Requirements:

1. The air induction system as installed shall prevent entrance of foreign matter during vehicle operation. Risk Level: High.
2. The air inlet shall be located to ensure that no water entry shall occur. Risk Level: Medium.
3. The air inlet shall be located in a low dust area to extend element life. Risk Level: Medium.
4. A resettable and graduated air filter restriction gauge shall be furnished. Risk Level: Low.
5. Pre-shaped tubing shall be used in the air induction system. Risk Level: Low.

FMTV Air Induction Systems Components:

1. Air Induction Sub-Assembly
2. Air Inlet

3. Air Filter Restriction Gauge
4. Air Induction Tubing

The air induction subsystem components used for this test case are illustrated as follows:

Figure 15: Air induction subsystem air inlet

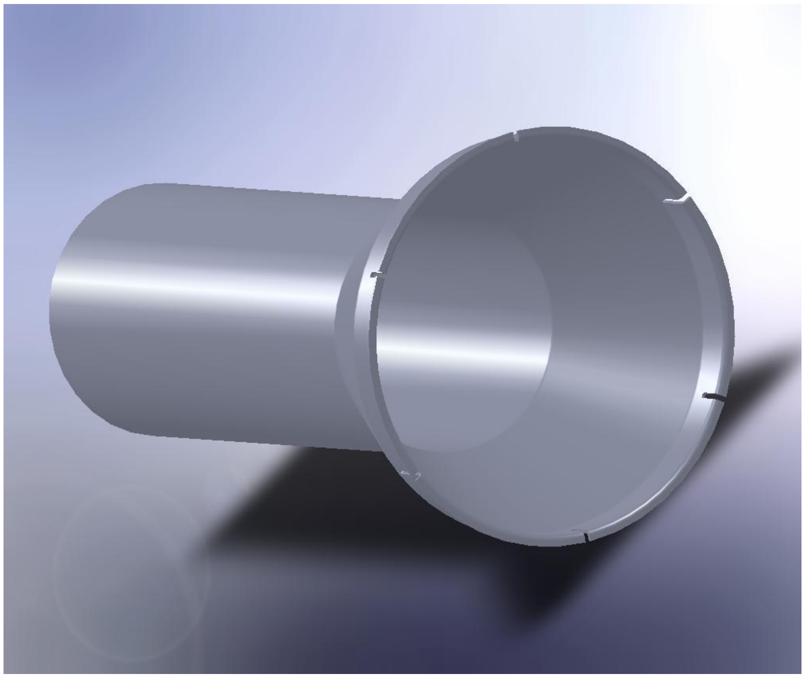


Figure 16: Air induction subsystem air filter restriction gauge

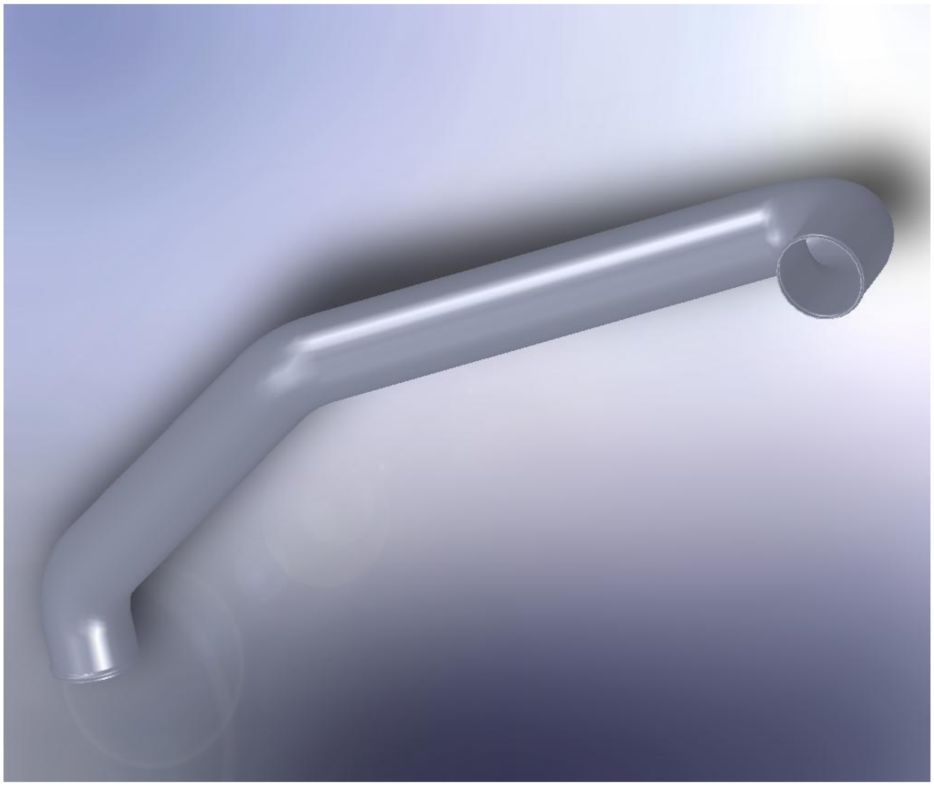


Figure 17: Air induction subsystem tubing

FMTV Air Induction Systems Test:

1. Engine Air Induction System Check

The Engine Air Induction System Check is a test that verifies the requirements were sufficiently satisfied.

Relationships between Components and Requirements:

1. Requirement 1 maps to the air induction assembly and consequently to the air inlet, air filter restriction gauge and the air induction tubing.
2. Requirement 2 maps to the air inlet.
3. Requirement 3 maps to air inlet.
4. Requirement 4 maps to air filter restriction gauge.
5. Requirement 5 maps to the air induction tubing.

Relationships between Requirements and Tests:

1. Requirement 1,2,3,4 & 5 map to Test 1.

Relationships between Requirements:

1. Requirement 1 maps to 2,3,4. Requirements 2,3,4 were derived from requirement 1.
2. Requirement 2 maps to 3. The position of the air inlet is addressed by Requirements 2 and 3. Therefore, any change in the air inlet position to satisfy Requirement 2 would also affect Requirement 3.
3. Requirement 3 maps to 2. The position of the air inlet is addressed by Requirements 2 and 3. Therefore, any change in the air inlet position to satisfy Requirement 3 would also affect Requirement 2.

Relationships between Components:

1. The air induction subassembly maps to the air inlet, air filter restriction gauge and the air induction tubing.

Implement the example problem in the software

As previously stated, the FMTV example problem is implemented in two software tools. While the specific implementation approach is dependent on the software tool used, the example problem is mapped to the eight capabilities in a uniform manner. First, ***refinement*** is demonstrated by first creating a "master" requirement and then representing the associated requirements that further define the requirements. ***Requirement history*** is demonstrated by modifying a previously modeled requirement. To test requirement ***satisfaction***, requirements and physical systems are modeled in the software and relationships between specific requirements and components are established. ***Input validation*** is evaluated by adding a nonsensical requirement to the project. The nonsensical requirement was created by interchanging parts of speech in the sentence. An adjective ('operable') was used as the subject, a noun ('elephants') was used as the verb and a verb ('accelerate') was used as the object of the sentence. The following nonsensical requirement follows.

1. The operable shall not elephants into the accelerate. Risk Level: High.

To ***verify*** requirements, corresponding verification tests are created and mapped to the requirements. Requirements ***prioritization*** is evaluated for all requirements in the example problem by assigning a high, medium or low priority. ***Coupling*** is evaluated by capturing the relationships requirement affect each other. Finally, requirement ***view restrictions*** are demonstrated by creating two user profiles and selectively filtering the modeled requirements in two views: the entire cooling system view of all requirements and a fluid system view, showing only requirements that affect the fluid of the cooling system.

IBM/Telelogic Doors

DOORS is a requirements management tool from IBM/Telelogic. This software enables requirements and other product information to be modeled and shared using a centralized repository. The DOORS interface is similar to a traditional word processing and spreadsheet program, allowing requirements documents to be published in a semi-formal manner. Design projects are modeled using modules to organize product information. These modules are used to organize data according to types such as functional/nonfunctional requirements, user requirements, system architecture or even smaller subsystems. Instead of compiling this information in one single document, this information is displayed in multiple smaller documents within the larger database. A module named "Air Induction System Requirements" is created. The modules serve as containers for the associated product information such as functional/nonfunctional requirements, user requirements, system architecture and tests (see Figure 18).

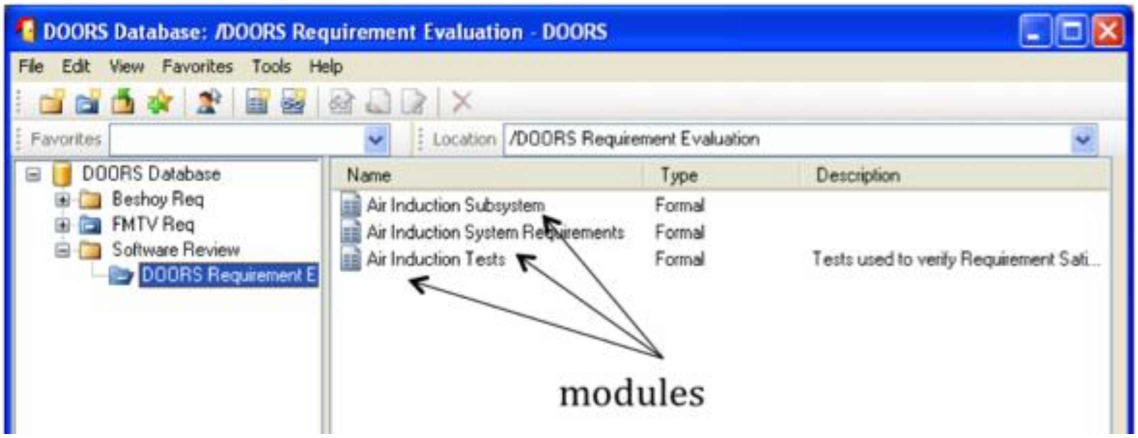


Figure 18: Initial window showing DOORS database and project modules for FMTV air induction subsystem.

First, a project is created in the DOORS Database for the demonstration example. In the current database there are three projects that can be accessed by engineering designers. For this project three modules are created for capturing Air Induction Subsystem (physical), Air Induction System Requirements, and Air Induction Test. Each of the engineering requirements are then created in the module and tracked using a unique ID. A requirement is modeled in DOORS by creating a new object in the requirements modules (Figure 19).

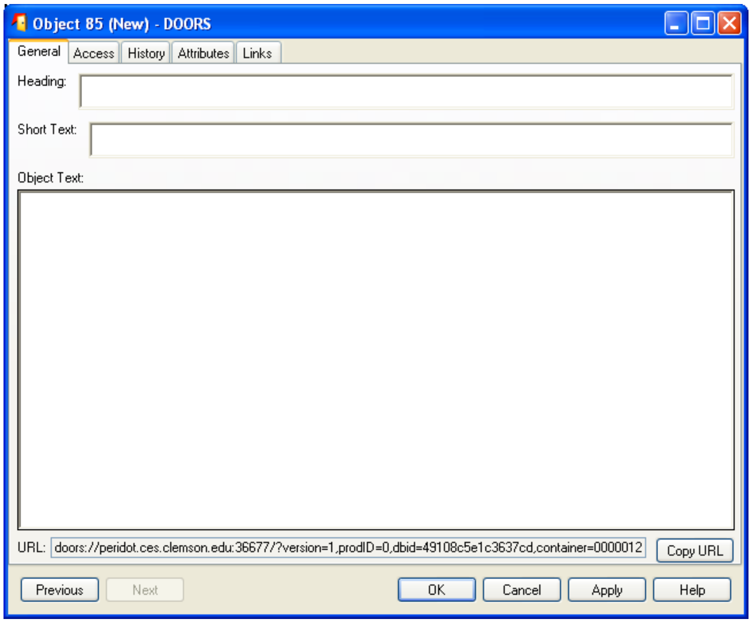


Figure 19: Screenshot of requirements object properties

Entered requirement information is shown in Figure 20.

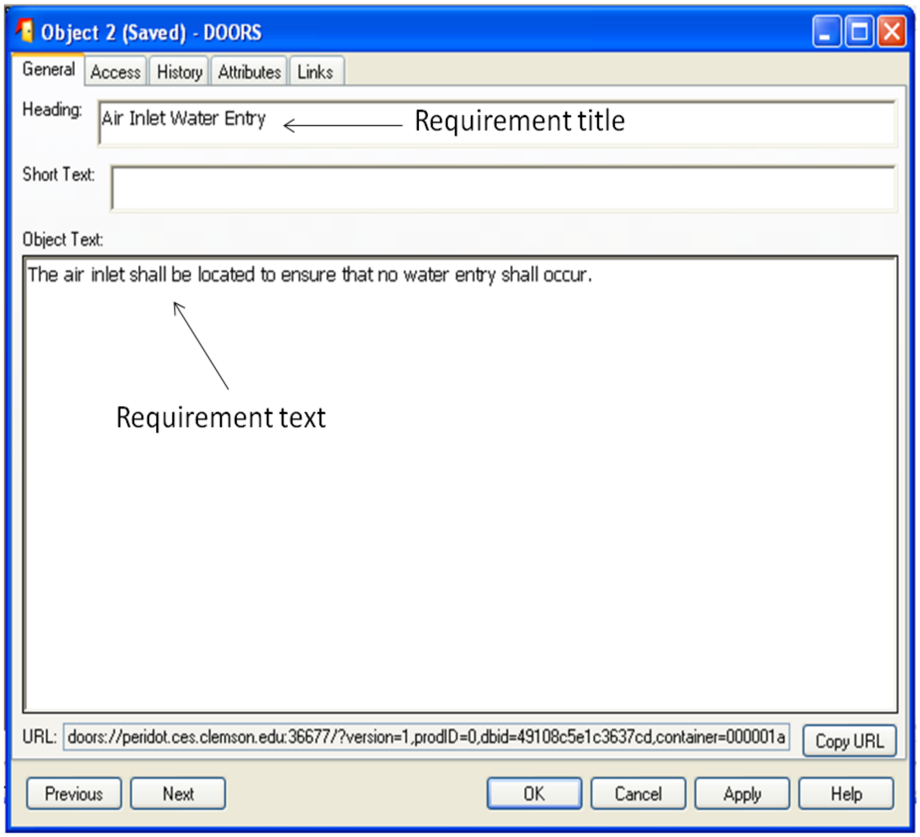


Figure 20: Screenshot of entered information in the requirements object properties

The "Heading" is used as the title of the requirement and the "Object Text" is used to display the complete requirement text. This process is repeated for all requirements that must be modeled.

Relationships between requirements and other information entities are modeled using links. Links are created as follows: a link is created from the starting entity as shown in Figure 21. The entity is selected by right-clicking and using the "Start Link" selection. This creates a link from that particular entity.

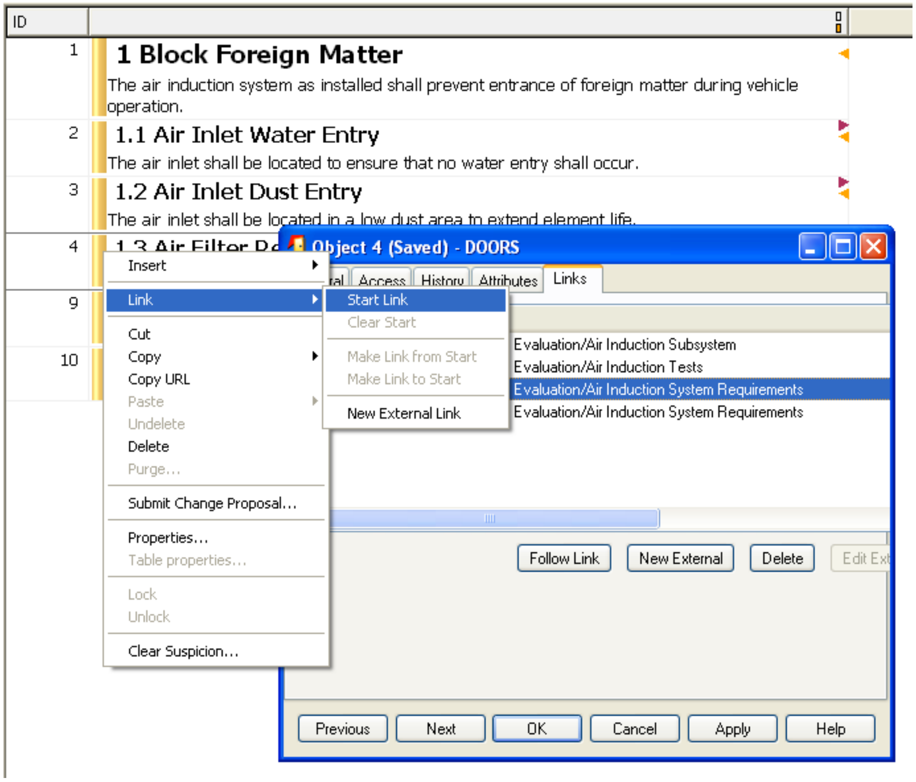


Figure 21: Link origin creation between two requirements

The relationship is completed when the link is terminated at the target entity. This is shown in Figure 22. The link is concluded by using the "Make Link from Start" selection.

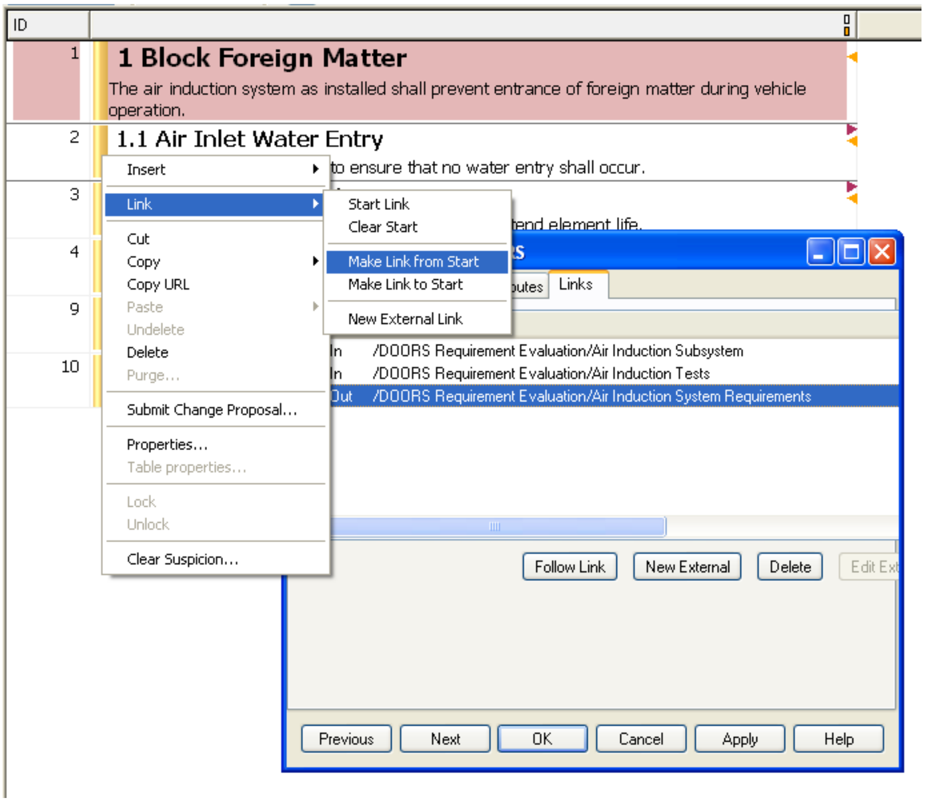


Figure 22: Completion of link between two requirements

In this case, the link was created from the “Block Foreign Matter” requirement to the “Air Inlet Water Entry” requirement. To view existing links, right-click the entity to investigate and select “Properties”. The links for the “Pre-shaped Tubing” requirement are illustrated in Figure 23 showing three ingoing links from three different modules: Air Induction Subsystems, Air Induction Tests and Air Induction System Requirements. One outgoing link exists to the module Air Induction System Requirements.

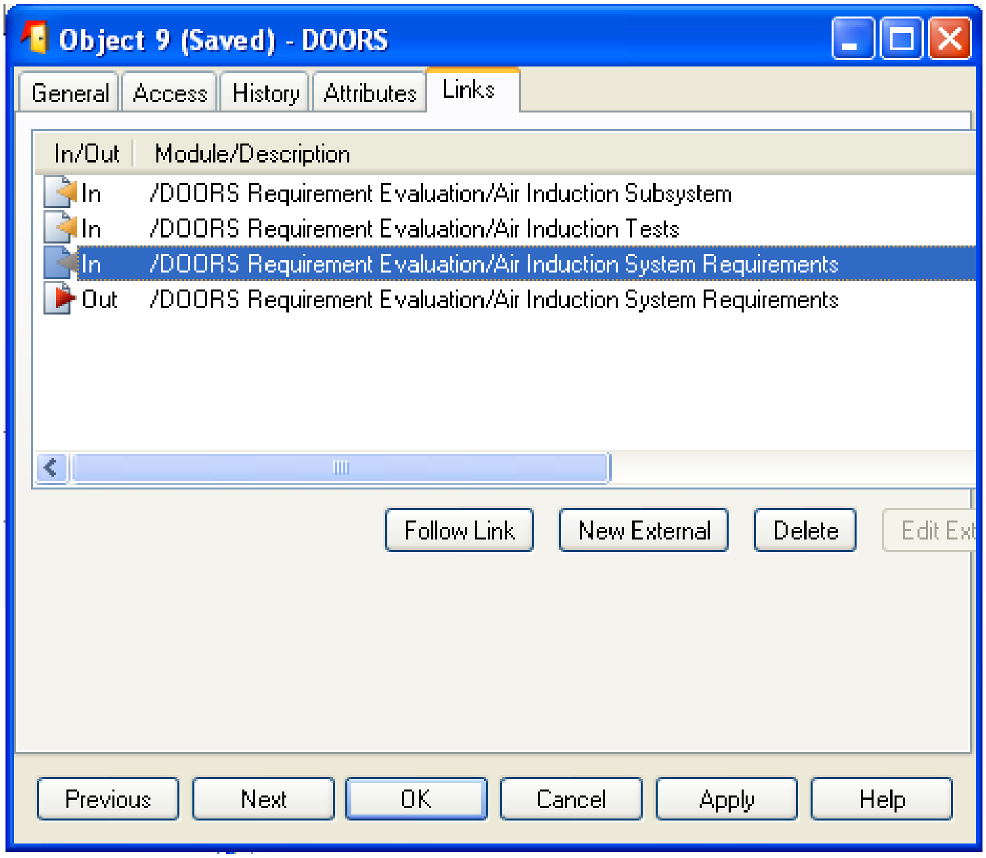


Figure 23: View of established links

Links may be created between different modules from a client in one module to the provider in another module. Outgoing links are denoted by red arrows and incoming links are denoted by orange arrows (see Figure 24).

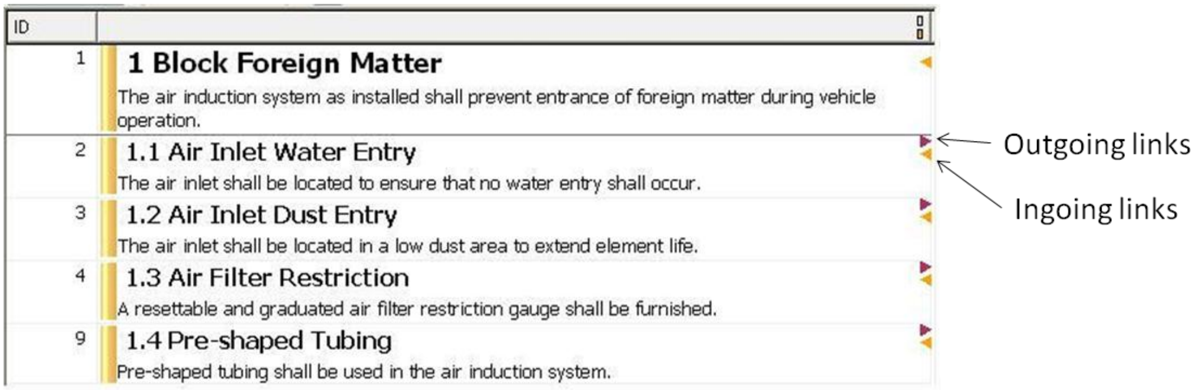


Figure 24: Air induction requirements in DOORS

The DOORS link utility provides a general approach for modeling several capabilities including satisfaction, verification, and coupling. While the link utility is flexible it leads to ambiguity in specifying relationships between different modules. For example, *coupling* and *refinement* are represented by the same arrows between requirements.

Refinement

Requirements *refinement* is modeled through an outline-based numbering scheme and using links. The refinement is shown in Figure 24 with the main requirement labeled as “1” and the sub requirements labeled as “1.x”. For example, the *“1. Block Foreign Matter”* requirement is refined through four additional requirements that specify the type of foreign matter and how it is blocked in requirements 1.1 through 1.4. This is shown textually in Figure 24 and graphically in Figure 25.

Figure 25: Requirements refinement – FMTV air induction subsystem requirements in DOORS.

The graphical representation of the requirements in Figure 25 is a tree structure with the parent requirement illustrated as a "tree branch" and the child requirements as the "leaf nodes". Refinement relationships are shown between requirements using links. These links are directional and denoted by triangles located on the right side of the cell. The beginning part of the link is denoted with a red triangle pointing out of the cell and the terminating link is denoted by a yellow-orange link pointing into the cell. The shortcoming of this type of relationship is that other types of requirement relationships (satisfaction and verification shown in Table 28) are represented with the same type of link.

History

The history of an engineering requirement is demonstrated by making several changes to Requirement 1.1: Air Inlet Water Entry. Referring to Figure 26, the requirement is changed five times (including the initial creation) in the door database. History is tracked in doors by recording the username, the edit session, edit date, and the specific modification(s) of the requirement. In Figure 26, the user (jmmclel) edited Object 2 multiple times (three times in session 1 and twice in session 3). The three editing instances in Session 1 involved creating the object, changing the text and changing the priority. In addition to capturing standard change data, it would be very useful to model the justification for the requirement change.

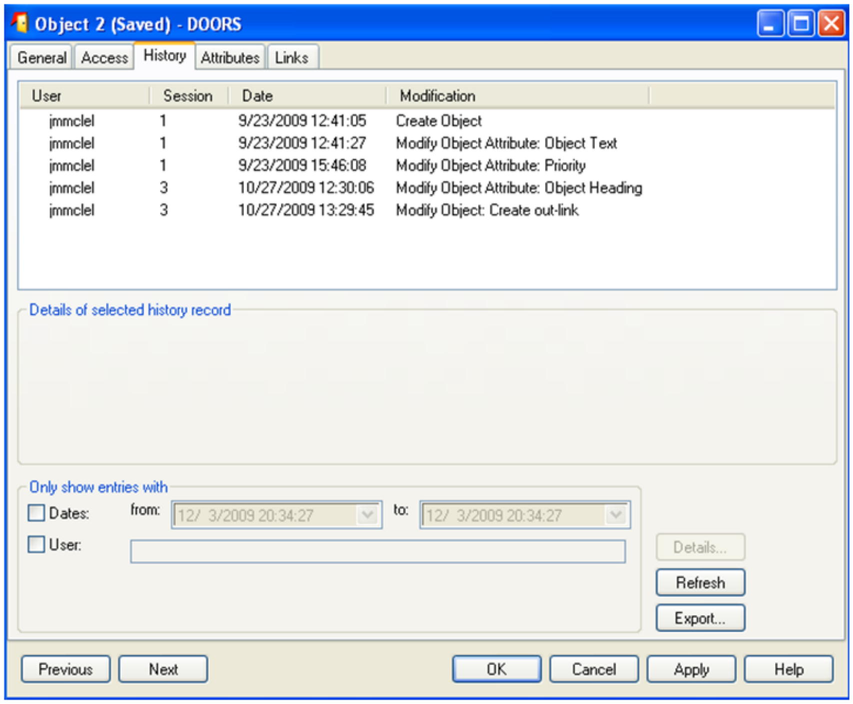


Figure 26: Requirement change history in DOORS.

Satisfaction

Requirements satisfaction is modeled by creating relationships between requirements and the physical design structure using links. In this design project satisfaction is created by linking the requirements defined in the Air Induction System Requirements module and physical entities in the Air Induction Subsystem module (see

Figure 27).

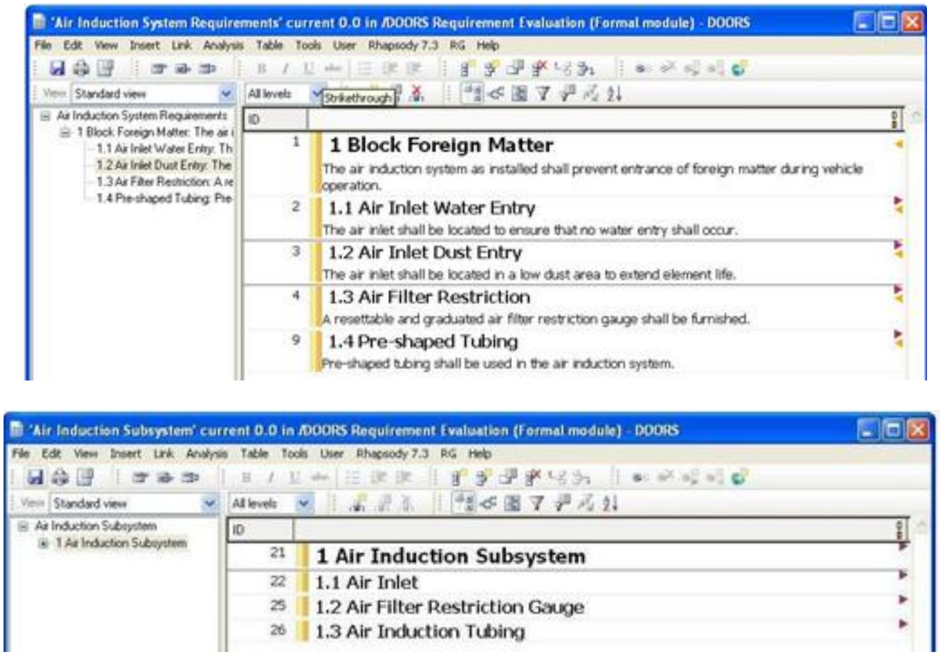


Figure 27: Requirements satisfaction – mapping air induction requirements and components.

Verification

Verification relationships are created by creating links between test cases and requirements. The directionality of the links shown in Figure 28 shows that the links were created from the Engine Air Induction System Check to the Air Induction Requirements in Figure 28.

The links modeled in DOORS are directional, thus identical relationships must be created from the Air Induction Subsystem module to the Air Induction System Requirements module (see Figure 28).

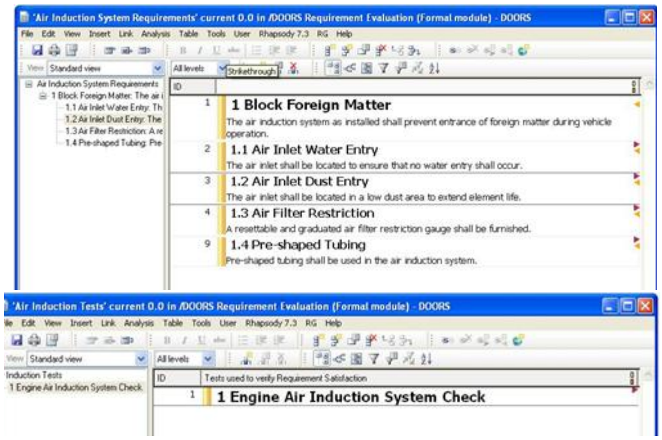


Figure 28: Requirements verification – mapping air induction requirements and test in DOORS representing using links.

The requirements (top of Figure 28) are linked, as indicated by the arrowheads to the physical system (bottom of Figure 28). However, the type and target of the link is not represented in the graphical window. As previously noted, the links are directional thus requiring explicit relationships to be created twice. Further, the ambiguity of the links does not enable the designer to distinguish between the type of relationship or target of the relationship.

Coupling

Referring back to Figure 21, DOORS allows couplings to be shown between requirements by linkings created inside the requirements module.

Prioritization

Requirements prioritization is implemented in DOORS by creating a user-defined priority tag and assigning a value of low, medium, or high. The air induction requirements are modeled in

Figure 29. Requirements documents can be sorted and filtered based on specified attribute values.

ID		Priority
1	**1 Block Foreign Matter** The air induction system as installed shall prevent entrance of foreign matter during vehicle operation.	High
2	**1.1 Air Inlet Water Entry** The air inlet shall be located to ensure that no water entry shall occur.	Medium
3	**1.2 Air Inlet Dust Entry** The air inlet shall be located in a low dust area to extend element life.	Medium
4	**1.3 Air Filter Restriction** A resettable and graduated air filter restriction gauge shall be furnished.	Low
9	**1.4 Pre-shaped Tubing** Pre-shaped tubing shall be used in the air induction system.	Low

Figure 29: Requirements prioritization – assigning levels of important to air induction requirements.

Input Validation

Input validation is not supported in DOORS. This capability was tested by adding the nonsensical requirement to the requirement module (see Figure 30). This nonsense requirement can be related to other requirements and components regardless of the content of the requirement. To ensure input validation, DOORS would have to have a vocabulary for each part of speech and ensure that the entered word is a part of the vocabulary.

Figure 30: Requirements input validation – addition of nonsensical requirement to air induction project.

View Restrictions

Finally, requirements view restrictions are based on user access (see Figure 31). For each requirement it is possible to grant controlled access for each requirement. As shown in Figure 31, each object created in DOORS has access rights that are granted to the user. These access rights can be inherited from the parent object or have completely different access rights independent of the parent object.

For example, the access rights for Object 3 are granted to user *bmorkos* for reading only, no editing possible. Access rights for *jmmclel* are granted for reading (R), modifying (M), create (C), delete (D) and administrator (A). Access rights for everyone else are denied.

Figure 31: Requirement view restriction in DOORS.

DOORS was found to successfully implement all of the requirement capabilities except for input validation. DOORS has some shortcomings since it could not differentiate relationships between requirement to component relationships, requirement to requirement relationships, component to component relationships and requirement to test relationships.

MagicDraw + SysML Plug-in

NoMagic MagicDraw+SysML is a modeling environment and plug-in for authoring SysML based representations. MagicDraw+SysML uses the requirement, block diagram and other diagrams (activity, use case, composite structure, etc) included in the SysML framework. This paper utilizes only the requirement and block diagrams.

To implement the demonstration example in MagicDraw, the requirements were created in a requirement diagram. A requirement diagram is a visual aid showing existing requirements and relationship types between them [24]. Creating requirements involves creating an empty requirements diagram and adding requirements. This is shown in Figure 32.

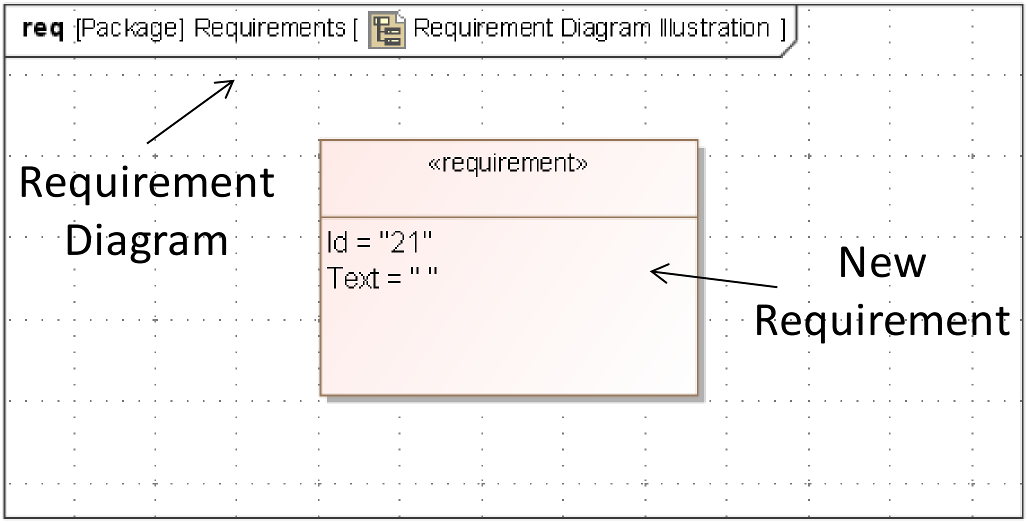


Figure 32: New SysML requirement diagram

If a new requirement is created, a requirement ID is automatically generated, as in Figure 32. To edit the requirement, the desired requirement is selected and a window of requirement options is shown. This window is illustrated in Figure 33. For the purposes of this study, only the requirement name, text, ID, and priority attributes are used.

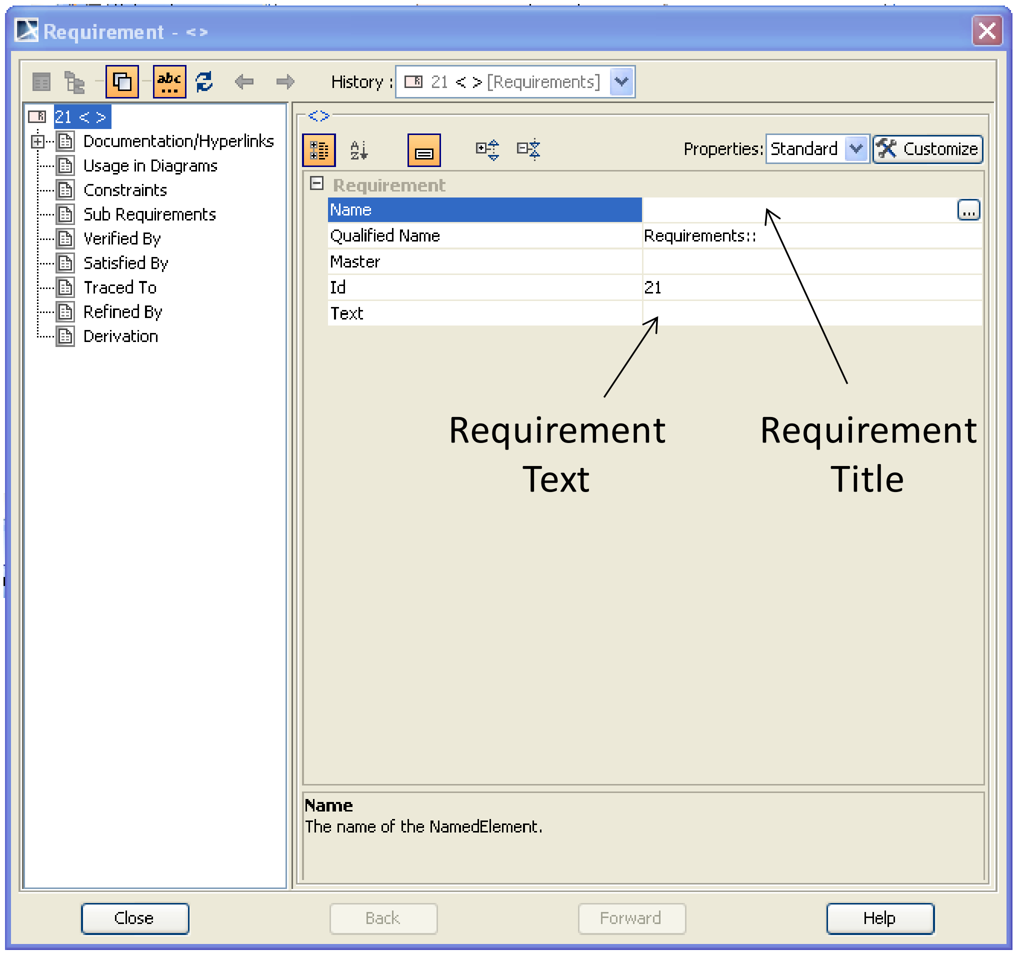


Figure 33: SysML requirement window

The requirement and test measure implementation in MagicDraw were both implemented in a requirement diagram as displayed in Figure 35. Relationships can be created between requirements by using standard or customized relationship stereotypes [35]. To create a requirement to requirement relationship, select the desired type of relationship and select the two requirements to be related. Selecting the new relationship line allows the relationship properties to be shown. This is shown with the "DeriveReqt" relationship in Figure 34.

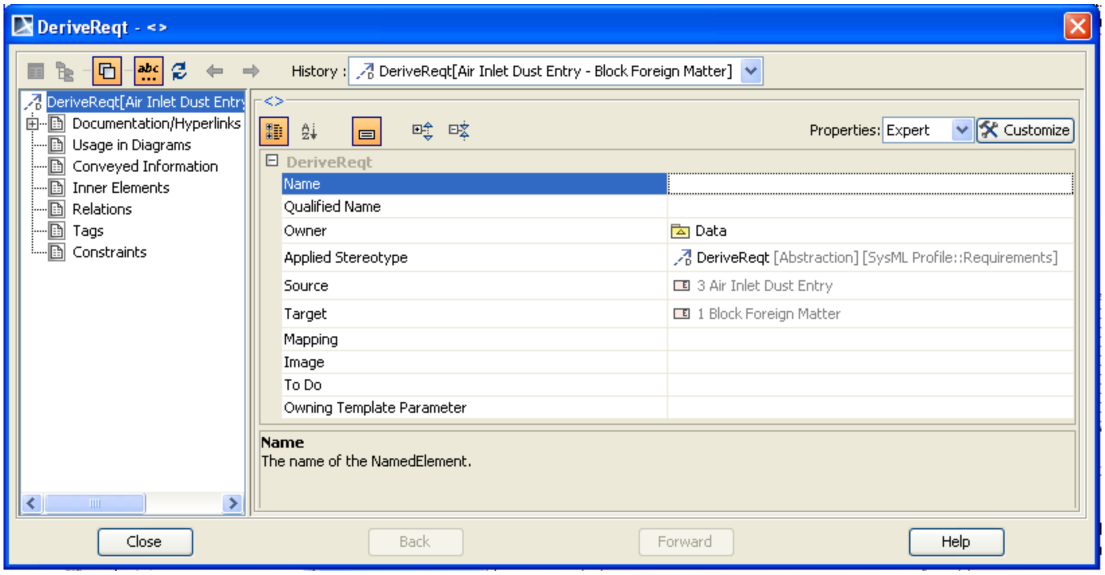


Figure 34: "DeriveReqt" properties

In this figure, the "Source" "3 Air Inlet Dust Entry" is the requirement derived from the "Target" "1 Block Foreign Matter". SysML provides the ability to record, keep track of, update and create relationships between requirements. A complete requirements diagram is included in Figure 35.

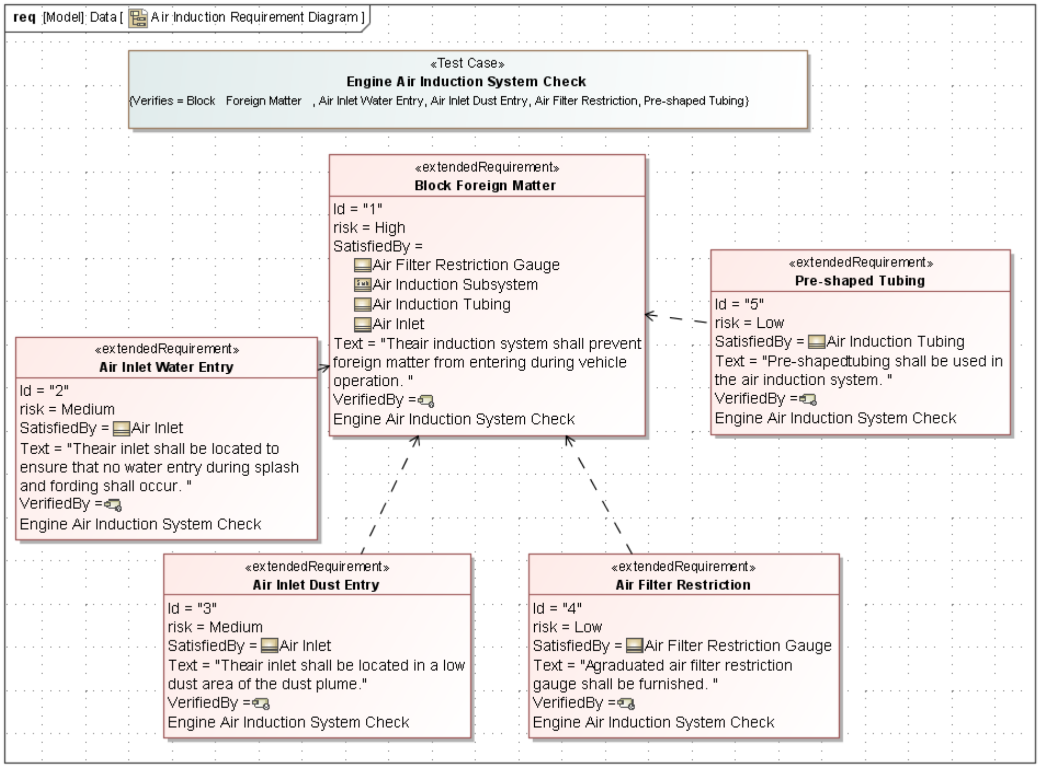


Figure 35: Complete SysML requirement diagram

The above figure shows a “SatisfiedBy” and a “VerifiedBy” relationship that has not been discussed yet but will be after the subsequent introduction and discussion of SysML “Blocks”.

The physical hierarchy is modeled using “blocks”. According to Weilkiens, blocks “describe parts of the structure of a related system” [35]. A block diagram is used to model the physical hierarchy. A new block and block diagram are illustrated in Figure 36.

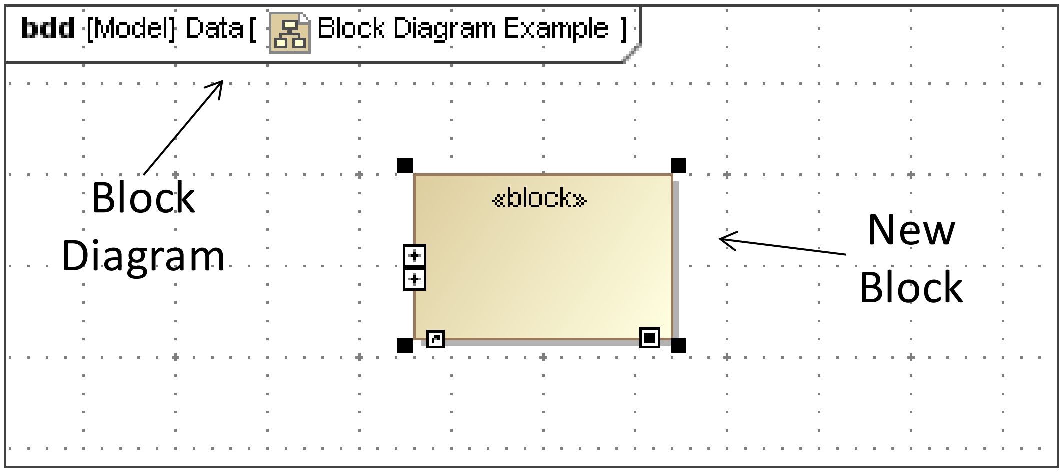


Figure 36: New SysML block diagram

Notice that blocks do not have ID's like requirements. Blocks can be organized into a hierarchy to illustrate the system structure. An example block diagram of the FMTV Air Induction Subsystem is included in the following diagram.

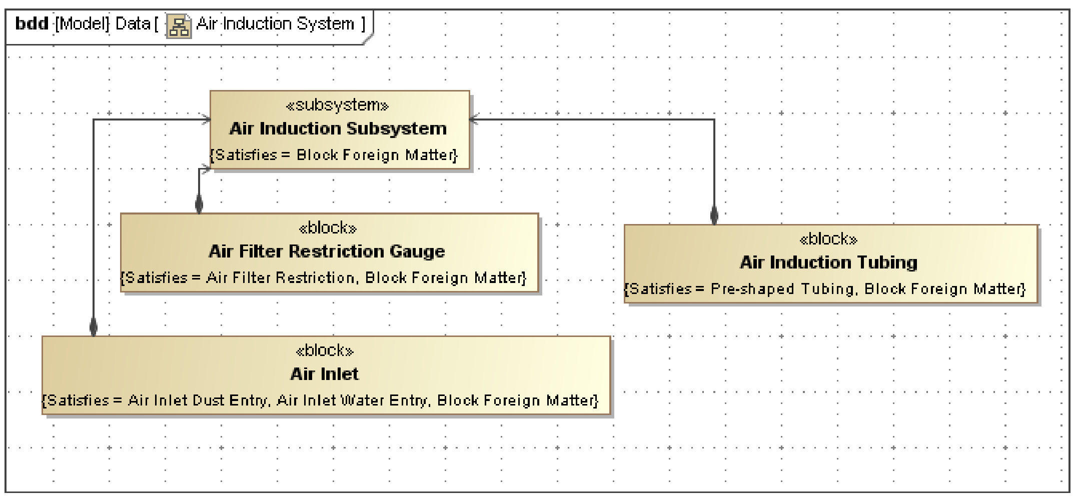


Figure 37: Air induction subsystem block diagram

In Figure 37, the air induction subsystem is related through a relationship called directed composition. Directed composition is used to illustrate a part/whole hierarchy [35]. Of particular note in this figure is the <subsystem> "Air Induction Subsystem". The <subsystem> is the larger structure that contains the blocks (or components).

To relate requirements and components, the "SatisfiedBy" relationship is used. To create these relationships, select the desired requirement, right-click and select "specification". Under the "tags" selection there are options for adding blocks that satisfy the requirements and adding tests that verify that the requirements are satisfied. Requirements are related to tests using the "VerifiedBy" relationship. This is shown in Figure 38.

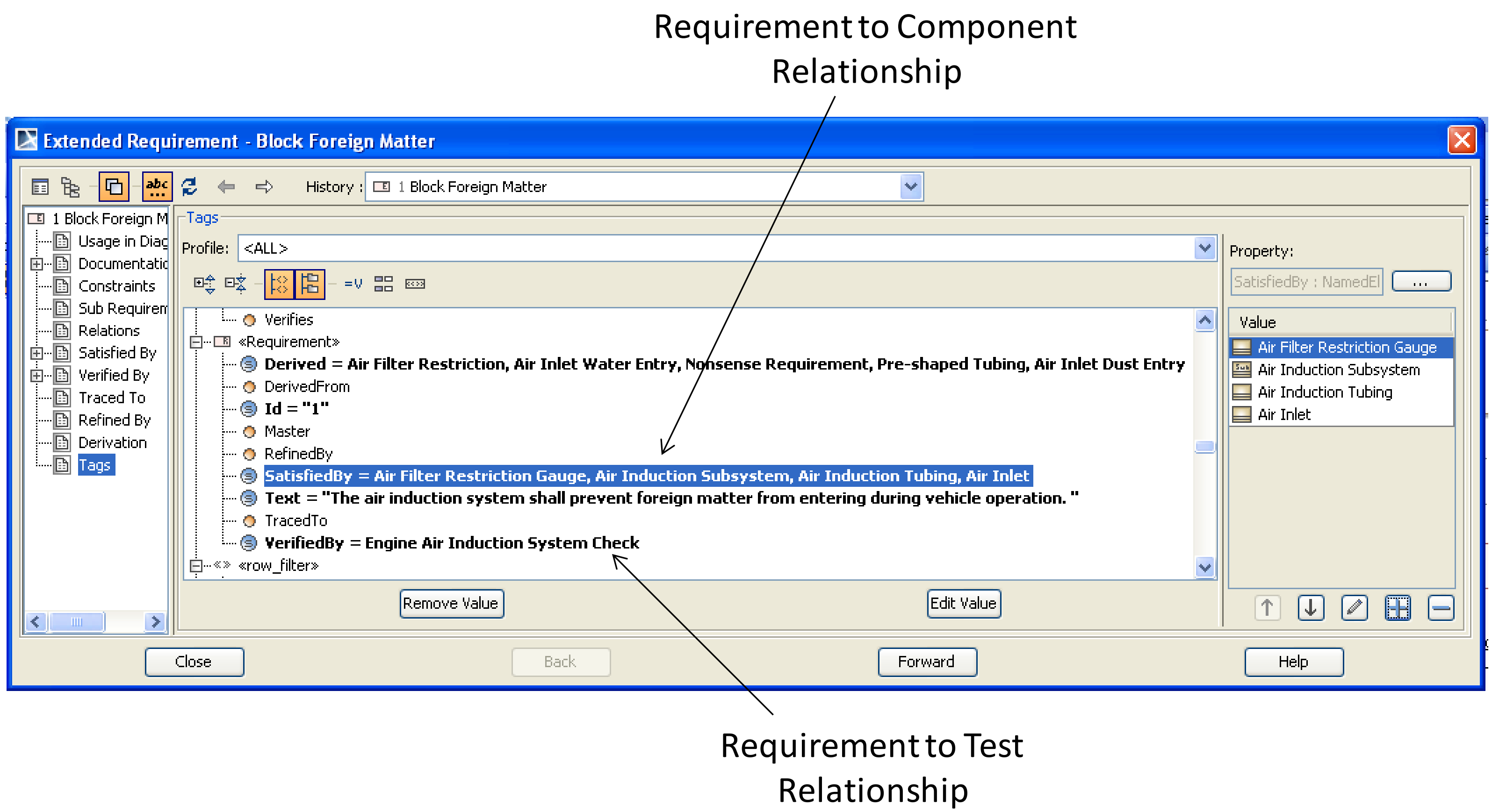


Figure 38: Creation of requirement to component and requirement to test relationships

The requirements and test measures were both implemented in a requirement diagram in Figure 35.

Refinement

Refinement is shown in Figure 35 using the “DeriveReq” relationship. Although not labeled as such in the figure, the “DeriveReq” relationship is displayed as the connecting arrows between requirements. In Figure 35, the requirements Air Inlet Water Protection, Air Inlet Dust Protection”, “Air Filter Restriction” and “Pre-Shaped Tubing” are all derived from the “Block Foreign Matter” requirement. The arrow direction of the “DeriveReq” relationship points from the derived requirement to the source requirement.

History

MagicDraw Teamwork Server can record the evolution of a requirement including who and how it was changed [44]. For this project, MagicDraw Standard Edition was used and was not integrated with the Teamwork Server.

Satisfaction

The system architecture is created in a block definition diagram in Figure 39. The hierarchy was defined using the SysML “composition” relationship (a subset of association and aggregation relationships) between blocks. The diamond at the base of the relationship shows the block that is composed of the other blocks at the end of that relationship. In this case the Air Induction Subassembly is composed of the components Air Inlet, Air Filter Restriction Gauge and the Air Induction Tubing.

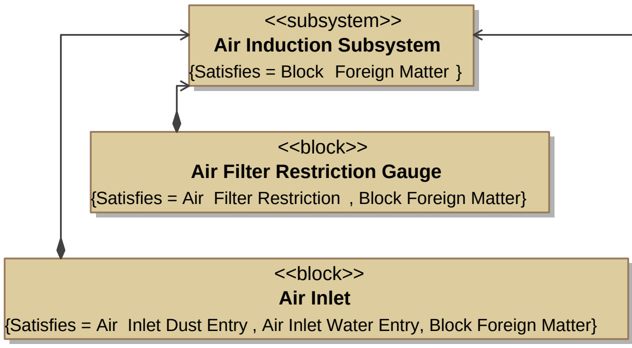


Figure 39: SysML air induction subsystem and components.

In Figure 35, a satisfaction relationship using the relational type "SatisfiedBy" is created between each requirement and its corresponding part(s) of the system architecture that satisfies it. This is included in the view of the requirement showing the block and block name that the requirement is satisfied by. For instance, the requirement Air Induction System is satisfied by the subsystem Air Induction Sub-Assembly. In Figure 39, a satisfaction relationship using the relational type "Satisfies" was created between the system architecture and the corresponding requirement. In summary, SysML has two relationships for showing satisfaction. One is "SatisfiedBy" and refers the requirement to the component(s) and the "Satisfies" relationship refers the component to the requirement(s).

Verification

Verification is shown in Figure 35 with Air Induction Subsystem Requirements mapped to the Engine Air Induction System Check Test Case. SysML has two directional relationships between requirements and test cases. The first relationship, "VerifiedBy", is directed from the requirement to the test case. This verification relationship can be viewed within each requirement. The second relationship, "Verifies", is directed from the test case to the requirement. This verification relationship can be viewed within each test case.

Coupling

Requirement coupling is shown in MagicDraw by using several different types of relationships. In this case, the “DeriveReq” relationship was used and is shown by the directional arrows from the four components Air Inlet Water Protection, Air Inlet Dust Protection, Air Filter Restriction and Pre-Shaped Tubing. The “DeriveReq” relationship not only establishes the requirement hierarchy used in refinement, but shows the inherent coupling that exists in hierarchies. MagicDraw can display relationships between entities in the same domain (Requirements to Requirements) or different domains (Requirements to Components) using relational matrices. Requirement coupling is shown in Figure 40. A requirement is coupled to other requirements if a relational arrow is connecting them. Requirements that are not connected by relational arrows (like the Air Inlet Water Entry and Preshaped Tubing in Figure 35) are not coupled. Relational matrices in MagicDraw show the directionality of the relationship from parent requirements to children requirements. Notice that the table in Figure 40 shows this.

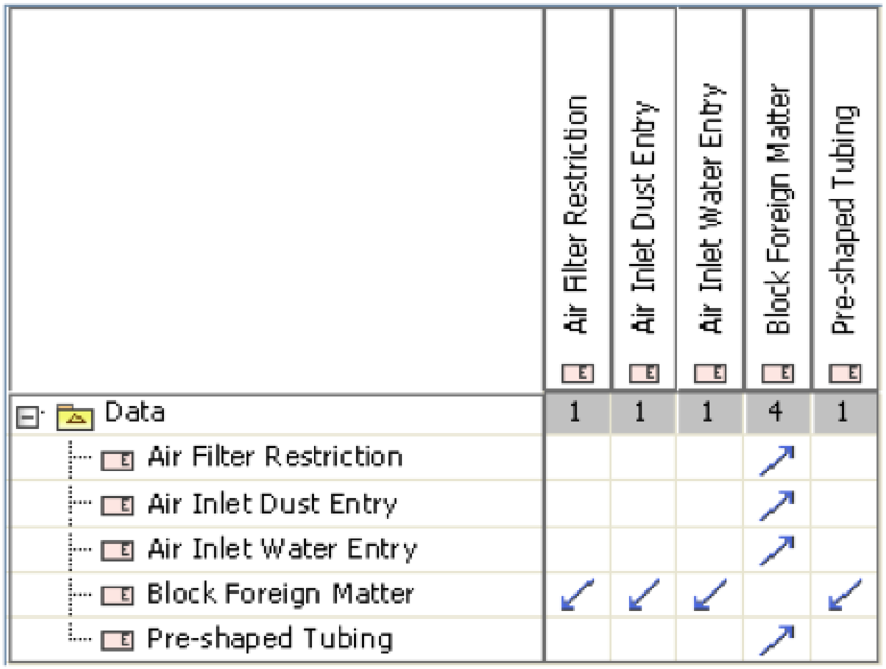


Figure 40: Requirements coupling as modeled in MagicDraw using the derived requirement relationship.

The row requirement Block Foreign Matter is related to the other requirements in that the other requirements are derived from it. Hence the arrow direction pointing from the children requirements back to the parent requirement. This is shown in the column also with the other requirements pointing to the parent requirement.

Prioritization

MagicDraw can handle prioritizing (in the MagicDraw Standard Edition) by using the risk attribute attached to the requirement. This can be seen in Figure 35 with the risk being assigned a "low, medium or high" value. Other more comprehensive MagicDraw licenses include a separate requirement attribute called "priority", but the "risk" attribute suffices in this case. We treat risk in such a way that an increased risk would entail a higher priority.

Input Validation

As can be seen from Figure 41, a nonsensical requirement can be added, coupled to other requirements, satisfy other components and be verified by other tests just as a valid requirement can. SysML lacks the capability for requirement input validation because requirements are modeled as text-based representations [24].

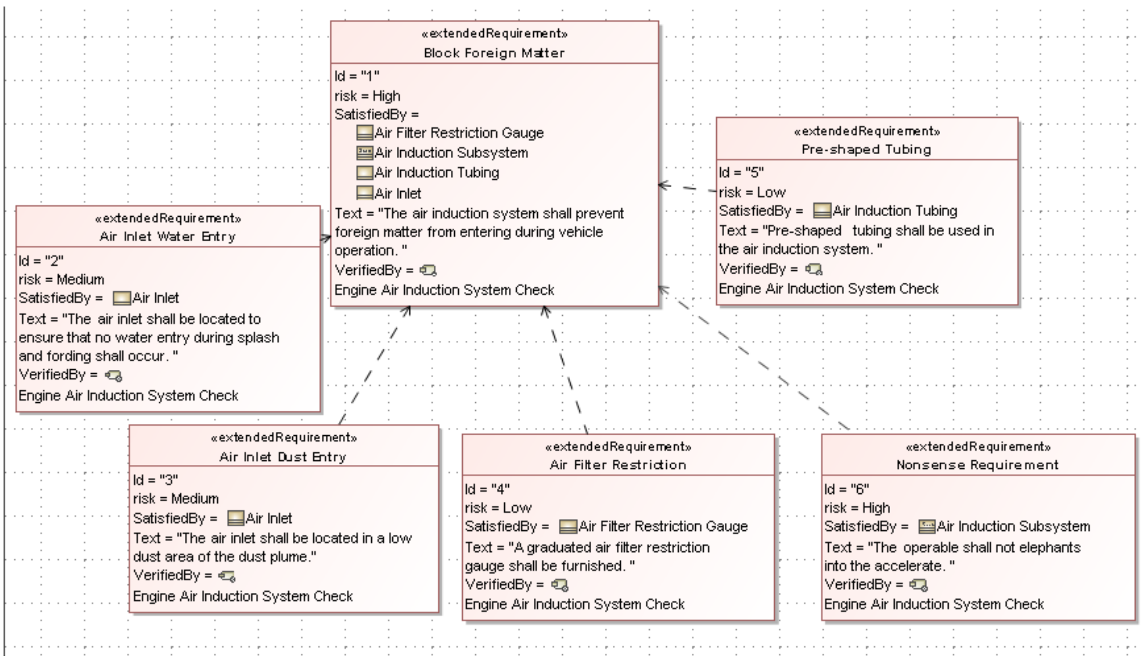


Figure 41: Air induction requirement diagram with nonsensical requirement.

View Restrictions

To create different views of requirements, the user can create different requirement diagrams and simply "drag" requirements from the existing requirements list to create different views. A dry-particle air induction view including the requirements applicable to this view is shown in Figure 42.

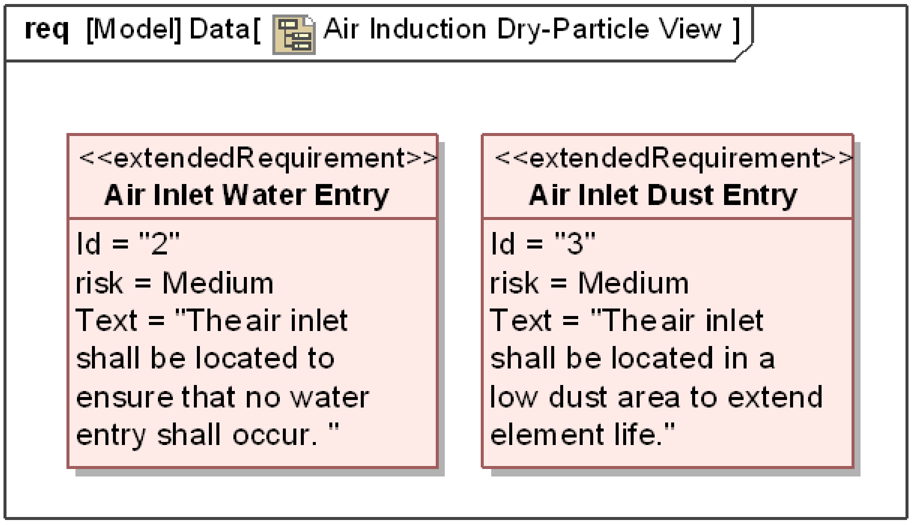


Figure 42: Dry-particle air induction view.

Evaluate the benefits and opportunities of the software

The observations and finding from the two implementations of the FMTV example problem in DOORS and MagicDraw are summarized against the eight requirements capabilities in Table 2. The main difference between MagicDraw SysML and DOORS is that DOORS is spreadsheet oriented, as can be seen from the cell approach to IDs, requirement name, text and attributes. MagicDraw is oriented according to block-like objects arranged in diagrams. Unlike DOORS which uses the same type of diagram which can be populated with requirements or whatever the user desires, MagicDraw has different types of diagrams to be used for requirements and system components. MagicDraw is more of a visual aid as custom diagrams can be made and the contained blocks (requirements, components, tests) can be rearranged inside the diagrams to the users desire.

Another large difference between the RM software DOORS and MagicDraw is that DOORS has only one generic relationship called the “link” that is used to relate all the domains to each other while MagicDraw has multiple types of relationships used to accomplish the same tasks.

DOORS uses the link relationship to relate requirements to themselves, to components and to test measures. MagicDraw has special relationship types for each of these tasks.

Both the DOORS and MagicDraw products support all of the capabilities except for input validation. In order to validate the input, SysML would have to include a sentence parser and a part of speech (POS) tagger and compare these against pre-existing vocabulary.

Requirement Capabilities	**DOORS**	**MagicDraw+SysML**
Refinement	The user can identify the decomposition schema of the requirements list using the hierarchal nature of DOORS.	A specific directional relationship can be created between requirements and their parent/child requirements.
History	The requirement change is recorded, including the date, by whom and the type of change made.	MagicDraw Team Server can record changes made to requirements. MagicDraw Standard Edition does not support requirement history documentation.
Satisfaction	Relationships between the system structure and the requirement are created using the "link" relationship.	Relationships between the system structure and the requirement are created using the "SatisfiedBy" relationship.
Verification	Relationships between the requirements and the test cases are created using the "link" relationship.	Relationships between the system structure and the requirement are created using the "VerifiedBy" relationship.
Coupling	Relationships between requirements are created using the "link" relationship.	Relationships between requirements are created using the "DeriveReq" relationship.
Prioritization	Requirements can be rated according to priority by creating an priority attribute. Requirement manipulation by sorting and filtering can then be done.	Includes a requirement "risk" attribute that can be treated as a priority attribute since higher risk requirements are of higher priority than lower risk requirements.
Input Validation	Input validation is not supported since there is no structure or standard to test requirements by.	Input validation is not supported since there is no structure or standard to test requirements by.
View Restriction	Viewing rights/limitations can be applied to users through their username privileges	Specialized requirement diagrams can be created with only certain requirements specified. It cannot create access rights for requirements.

Table 29: Evaluation of requirements management software versus requirement capabilities

The main difference between handling requirements and their related domains in DOORS and MagicDraw is that DOORS has only one generic relationship called the "link" that is used to relate all the domains while MagicDraw has multiple types of relationships used to accomplish the same tasks.

Both MagicDraw and DOORS allows for requirement decomposition. DOORS shows a "outline view" and a hierarchal tree structure view. MagicDraw allows for boxes (objects) connected by lines (relationships) to denote the decomposition. DOORS allows for recording of requirement history which is in scope with its requirements management uses. MagicDraw lacks the ability to record requirement history which is in scope with its system design uses. Both DOORS and MagicDraw can account for requirement satisfaction and verification, the primary difference being the type of relationships used to create the satisfaction and verification relationships. The two verification relationships seem to be extraneous. A directional verification in one direction necessitates a relationship in the other direction. If a requirement is verified by a test case, the test case verifies the requirement. The two satisfaction relationships seem to be extraneous. A directional satisfaction in one direction necessitates a relationship in the other direction. If a requirement is satisfied by a component, the component satisfies the requirement.

Based on the results of modeling DOORS and MagicDraw, it was concluded that MagicDraw should be used to implement the requirements since it uses different relationship types when relating different design domains.

Chapter 5. Introduction to Family of Medium Tactical Vehicles

The FMTV system is introduced in regard to the vehicles specific subsystems in this chapter.

Overview of FMTV

The FMTV is currently being produced by BAE Systems. Stewart & Stevenson was awarded the contract in 1991, successfully rebid the contract in 2003 to produce the FMTV until 2009 [45]. BAE Systems acquired Stewart & Stevenson in 2006. In 2009, Oshkosh won the contract for producing the FMTV through 2015 [46]. However, the ARMY TACOM owns the Technical Data Package for the FMTV, thus the new 2009 contract will be a "build contract". The FMTV variants are summarized in Figure 43.

Figure 43: Family of medium tactical vehicles (FMTV) [46]

The FMTV is delivered in a 2.5 or 5.0 ton platform, with the key difference being the number of axles on the vehicle and rated engine horsepower. The 2.5 ton vehicle has 2 axles, whereas the 5 ton vehicle has three axles. Further, the rear frame of the vehicle is modular to contain any of the above configurations including tractor trailer, cargo flatbed, wrecker (tow-truck), van, and dump bed. The FMTV is designed to be the backbone for the Army's unit mobility and logistics support. This vehicle operates throughout the world in extreme weather conditions from -50°F to +120°F [47,48]. The FMTV serves as the basis for demonstrating the method developed in this research. The FMTV example is chosen for the following reasons:

1. Availability of requirements data
2. Access to geometric models and component information
3. Extensive use by the Army

In this research, three sub-systems that are common across all FMTV variants are analyzed: the Engine Cooling Subsystem, Chassis Subsystem, and the Cab Subsystem. Notice that these subsystems are shared by all FMTV variants. These subsystems were chosen because of:

1. The availability of subsystem requirements
2. The availability of specific component requirements within their respective subsystem

Other subsystems could have been used to implement the requirement analysis model, but they lacked requirements at either the subsystem or component levels. Since this model is requirement based and uses mass, the optimal situation is to have both subsystem and component requirements to analyze.

The subsystems discussed in this thesis of the 2.5 ton FMTV truck type are described in the following sections.

Modeling FMTV Requirements and Physical Components

FMTV Cooling Subsystem

The FMTV cooling system is designed to maintain engine temperatures of a Caterpillar C7 power plant, a heavy duty diesel, 6-cylinder, electronically controlled, fuel-injected turbocharged and after cooled engine. The engine produces 275 hp (205 kW) at 2200 rpm displacement with 441 cu in (7.2 L). The engine torque is 860 lb-ft (1,166 Nm) at 1440 rpm. The cooling system is designed to maintain temperatures of the engine oil, engine coolant and transmission fluid. Cooling is accomplished by liquid-air cooling for the engine coolant and liquid-liquid cooling for the engine oil and transmission fluid. The engine coolant is used to cool the engine oil and transmission fluid (liquid-liquid cooling). The engine coolant is cooled by forcing air over coils (liquid-air cooling). The components included in the cooling subsystem are as follows:

- FMTV Cooling Subsystem
    - Heavy Duty Clamps
    - Coolant Hoses
    - Transmission Oil Cooler
    - Water Pump
    - Coolant Overflow Chamber
    - Auxiliary Oil Cooler
    - Charge Air Cooler
    - Seals
    - Centrifugal Cooling Fan Subsystem
        - Radiator Fan Bottom Shroud
        - Radiator Fan Top Shroud
        - Centrifugal Fan

            - Fan Clutch

FMTV Chassis Subsystem

The FMTV uses ArvinMeritor axles with Michelin XML 395/85R 20” all-terrain tires. The suspension system uses parabolic-tapered leaf springs with coil over hydraulic shock absorbers in the front with parabolic-tapered leaf spring with hydraulic shock absorbers and stabilizer bar in the rear. The components included in the chassis subsystem are as follows:

- Chassis Subsystem
    - Trailer Hitch
    - Rear Axle
    - Rear Axle Housing
    - Front Axle
    - Front Axle Housing
    - Main Beams Subsystem
    - Leaf Springs
    - Fifth Wheel
    - Tires
    - Winch

FMTV Cab Subsystem

The FMTV cab design is a three-man, ergonomically adjustable driver seat with steering power assist, recirculating ball storage 8 cu ft (2.4 cu m) and a three-point rubber isolator for the cab suspension. The cab subsystem included in this study is as follows:

- Cab Housing (sides and roof)

- Steering Wheel
- Instrument Panel
- Cab Floors

The next three chapters will discuss the implementation of the three FMTV subsystems with the requirement analysis method. Each chapter will discuss one subsystem. Chapter 6 will analyze the FMTV cooling subsystem, Chapter 7 will analyze the FMTV chassis subsystem and Chapter 8 will analyze the FMTV cab subsystem. The requirement analysis method will identify several requirements from each subsystem that can be altered to affect mass.

Chapter 6. Analysis of FMTV Engine Cooling Subsystem

The requirements modeling method developed in Chapter 3 is used to analyze the FMTV Engine Cooling, Chassis and Cab Subsystems. It is displayed for review purposes in Figure 44.

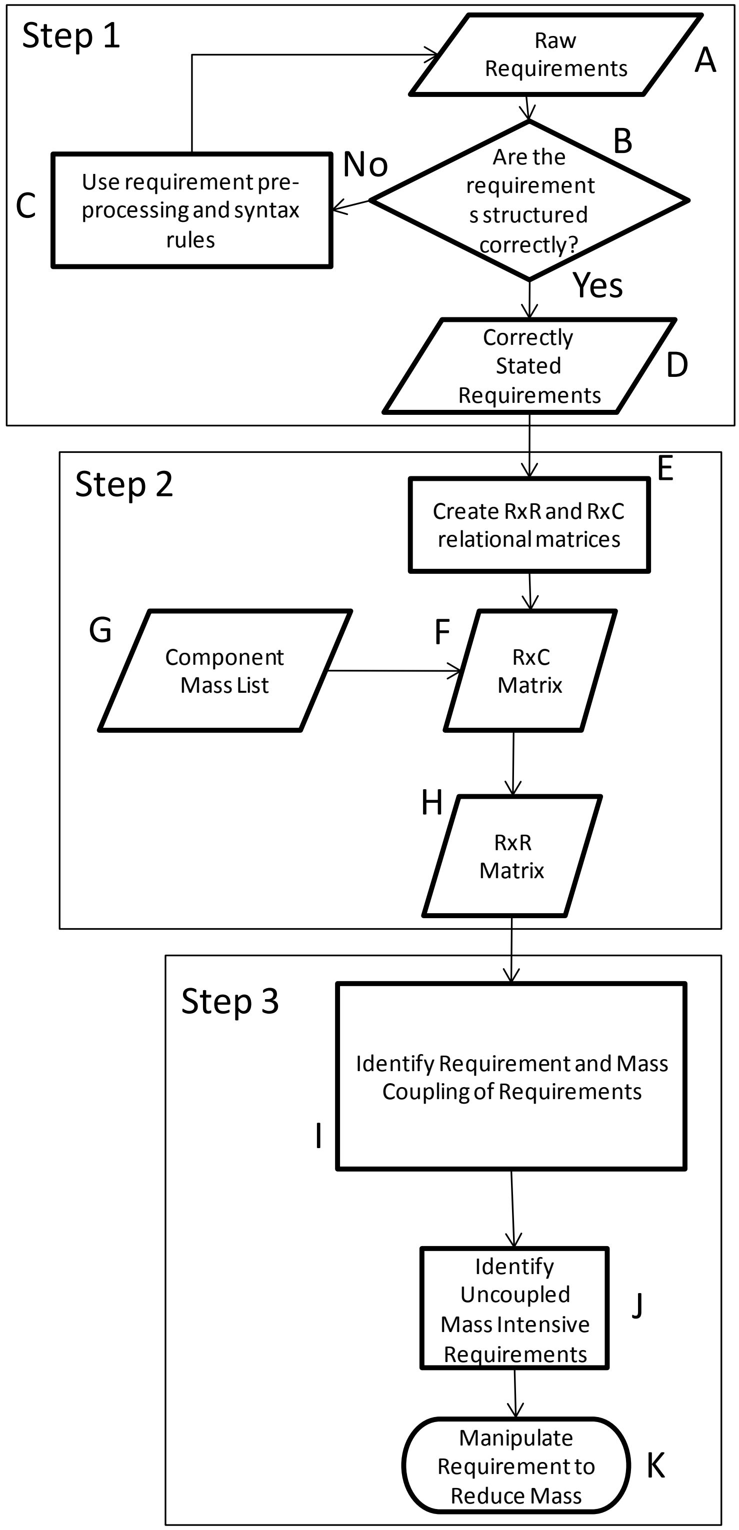


Figure 44: Requirement analysis flowchart

In this chapter the FMTV Cooling Subsystem will be discussed and is shown in Figure 45. The other subsystems will be discussed in subsequent chapters.

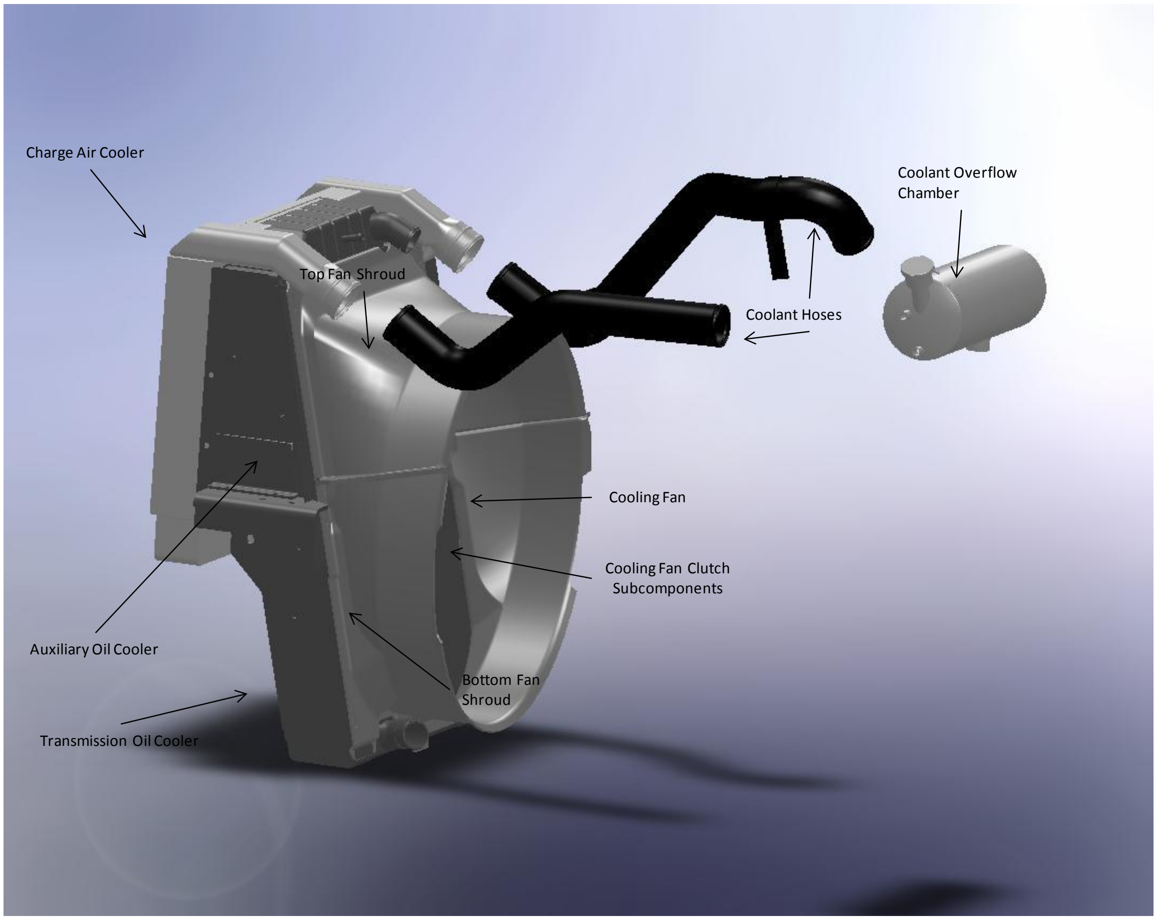


Figure 45: FMTV cooling subsystem

In Figure 45, the labels indicate the location of the components. Some components are not included in this figure, namely the cooling fan and the cooling fan clutch subcomponents.

The data for the analysis is obtained from several different sources including (1) ATPD2131F.1 and (2) CAD models. The mass reduction method for each subsystem is described step by step in the following sections.

Step 1: Acquire and process requirements

The unprocessed FMTV Cooling Subsystem Requirements used in the requirements analysis method are included in Table 30.

Table 30: Unprocessed FMTV cooling subsystem requirements

No.	Text
3.2.1.12	Any components exposed up to XX inches from the ground with the emergency CTIS setting in force, to include hoses, cables, lanyards, lines, tanks, valves, wires, cylinders, boxes, shall be shielded or able to withstand, going in forward or reverse, with no degradation of vehicle operation: the repeated impact of brush and tree branches; dry debris raised by cross country operation; soil scraping at XX mph.
3.3.1	All materials shall be new and unused.
3.3.3	Workmanship shall be of the highest grade consistent with the intention of this specification. Each vehicle shall have no evidence of cracks, dents, scratches, burrs, sharp edges, loose parts, foreign matter, or any other evidence of poor workmanship that shall render the vehicle unsuitable/unsafe for the purpose intended.
3.4.1	The fan clutch shall be such that, in the event of failure, the fan shall be constantly engaged.
3.4.1.2	The cooling system shall be capable of retention and recovery of XX% coolant overflow or have XX% expansion reserve capacity. The cooling system shall be capable of continuous de-aeration of XX cfm of air per cylinder at rated engine speed at any slope the vehicle is required to operate on. The system shall fill completely, with an automatic de-aeration feature to preclude air cavitation at any coolant fill rate up to the maximum fill rate. Maintain the specified component operating temperatures within the specified limits while operating continuously at full load and XX tractive effort to gross vehicle weight ratio (TE/GVW) while under the maximum conditions of XXo F for all models with the exception of the Expansible Van, LHS, Tractor and Wrecker which shall meet a minimum of XX TE/GVW while under maximum conditions of XXo F. Does not exceed temperature limits while operating at rated engine power. Meets the requirements after a drawdown of XX% of engine coolant. Specified fluid temperatures shall not exceed the lower of those for which the component manufacturer shall provide warranty, or the following: REFER TO PG 21. The radiator shall have a maximum of XX fins per cm and shall be located to minimize air side fouling. Heavy duty clamps shall be used, shall be clearly visible, located for ease of connection, and ensure positive sealing. The cooling system shall not be comprised of heat exchangers in series in areas prone to fouling.

After applying the pre-processing and syntax rules the following structured requirements in Table 31 were written.

Table 31: Structured FMTV cooling subsystem requirements

No.	Text
3.3.1b	All materials shall be new and unused.
3.3.3	Cooling system shall be of the highest grade of workmanship consistent with the intention of this specification.
3.4.1.2e	Cooling system shall fill completely with an automatic deaeration feature to preclude air cavitation at any coolant fill rate up to the maximum fill rate.
3.3.3a	Cooling system shall not have blemishes that shall render the vehicle unsuitable/unsafe for the purpose intended.
3.4.1.2q	Heat exchangers shall not be in series in areas prone to fouling.
3.4.1.2f	Cooling system shall maintain the specified component operating temperatures within the specified limits while operating continuously at full load and XX tractive effort to gross vehicle weight ratio (TE/GVW) while under the maximum conditions of XXo F for all models.
3.4.1.2h	Cooling system shall maintain component operating temperatures less than the temperature limits while operating at rated engine power.
3.4.1.2j	Cooling system shall maintain temperatures less than the lower of those for which the component manufacturer shall provide warranty, or the following: REFER TO PG 21 of ATPD2131f.1.
3.4.1.2i	Cooling system shall meet the requirements after a drawdown of XX% of engine coolant.
3.2.1.7c	Seals shall restrict the entrance of foreign matter into bearings which are exposed to contamination during these operations.
3.4.1.2o	Heavy duty clamps shall be located for ease of connection.
3.4.1.2m	Heavy duty clamps shall be used for the radiator.
3.4.1.2p	Heavy duty clamps shall seal completely.
3.4.1b	Fan clutch shall be engaged constantly in the event of failure.
3.2.1.12a.2	Fluid lines shall be protected by routing or placement in areas shielded by heavier components.
3.2.1.12b	Fragile components shall be protected from repeated impact of brush and tree branches, dry debris raised by cross country operation and soil scraping at XX mph while exposed up to XX inches from the ground with the emergency CTIS setting in force.
3.4.1.2n	Heavy duty clamps shall be clearly visible.
3.3.1.3	Nonmetal components shall not deteriorate due to mold, fungus, moisture, repeated exposure to bright sunlight, or use while stored in accordance with TM 9-2320-391-20, Section IV, Chapter 2-21.
3.4.1.2l	Radiator shall be located to minimize air side fouling.
3.4.1.2k	Radiator shall have a maximum of XX fins per cm.
3.4.4b	The transmission heat exchanger shall have a heat exchanger which does not rely on air flow over the transmission as recommended by the manufacturers.

These cooling subsystem requirements refer not only to the leaf node parts of the physical design (components) but to the branch as well (cooling system). Once the requirements have been standardized, the next step of obtaining component information is started.

Step 2: Map Requirements to Components

Once the Requirement Analysis Method was developed, it was implemented on three FMTV subsystems: the cooling subsystem, the chassis subsystem and the cab subsystem. To implement the requirement analysis method on the FMTV truck subsystems, the component masses for specific subsystems were acquired. The provided component files were created by the Army in Pro-Engineer modeling software. The only computer modeling software available to conduct this research was SolidWorks. Only a few of the components imported into SolidWorks were recognized as solids and analyzed for mass. Most of the shapes used to create the component in Pro-Engineer were imported as surfaces in SolidWorks and could not be knitted together. This kept the surface from being converted to a solid and a volume measurement could not be made. Consequently, the mass could not be calculated. An illustration of surface figures that could not be converted to solid figures for the FMTV engine block are represented with blue outlines in Figure 46.

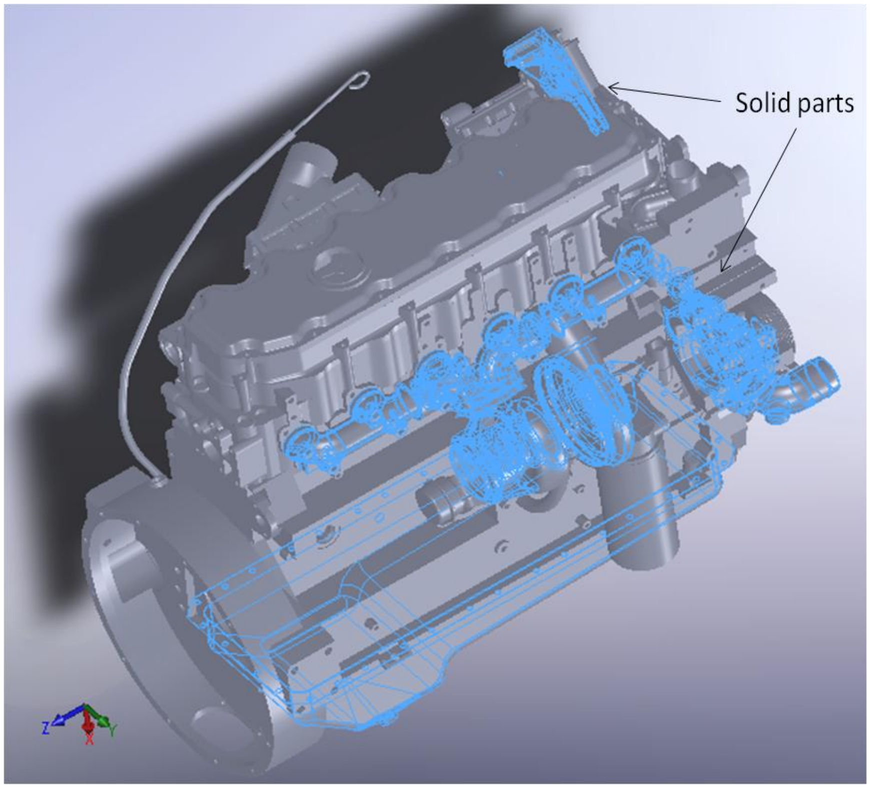


Figure 46: Surface structures for the FMTV engine block

An approximation for the volume was found by measuring the surface area as shown in Figure 47 and multiplying it by the "thickness" surface as illustrated in Figure 48. The blue plane in Figure 47 was the plane selected to have its surface area measured. In Figure 47, the width of the blue plane was found as the depth of the surface found in Figure 48.

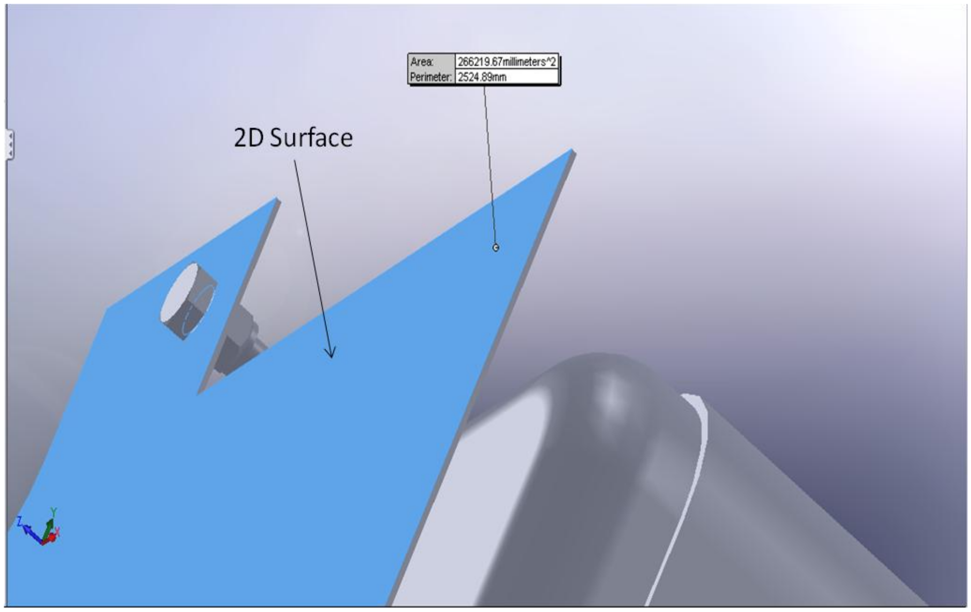


Figure 47: Solidworks surface area measurement

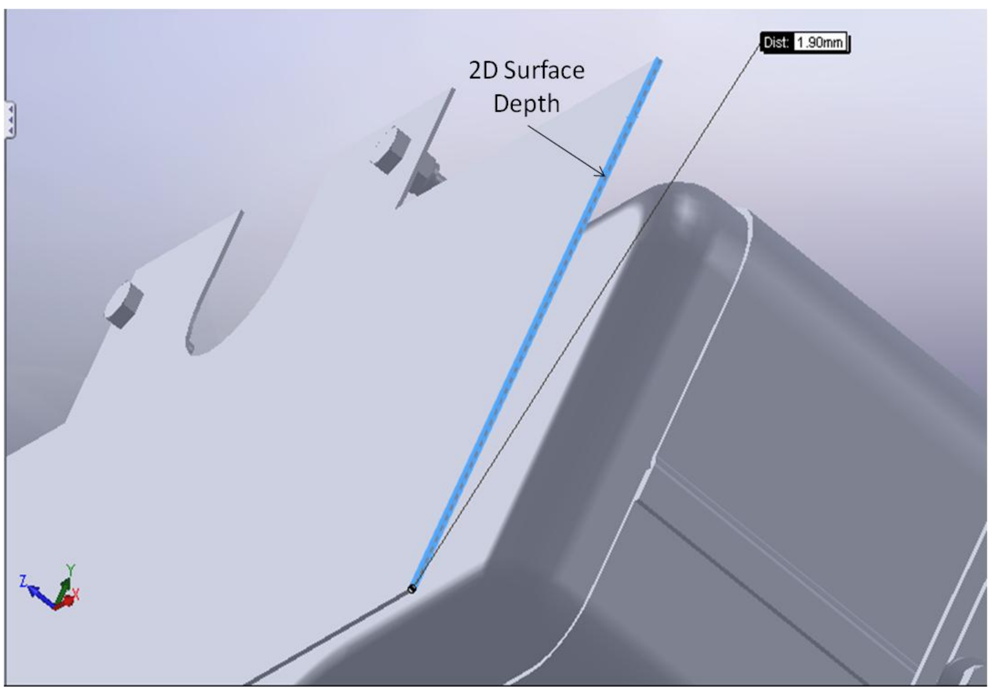


Figure 48: SolidWorks measurement of surface "depth"

This approach for calculating the mass works well for flat surfaces. However, curved surfaces present a problem. The inner surface of a curved surface does not have the same surface area as the outer curved surface. This problem is easily visualized by flattening an orange peel. Because the inner surface of the orange peel is not the same surface area as the outer surface, the orange peel tears or distorts in an attempt to compensate for the tension. In this case, the inside surface area of a curved surface was assumed to be the same as the outside surface. Thus, calculating the volume and mass of a curved shape of uniform width was approximated. This amount of error is very small and in our case negligible since the overall mass of the FMTV is much greater than the mass error due to approximation. For the Requirement Analysis Method, the exact mass isn't as necessary as the magnitude that each component has. Some of the subsystem components were not found to have a .prt file and thus comparable parts were found on the internet and their masses were used. These components are marked with an NA for not applicable since an image was not created in SolidWorks of that file. The FMTV engine cooling subsystem component list is shown in Table 32 and the component hierarchy modeled in SysML is shown in Figure 49.

Table 32: FMTV Engine Cooling Subsystem Component List

Component	Figure	Mass (kg)
FMTV Cooling Subsystem	Figure 64	65.08
Heavy Duty Clamps	NA	1.00
Coolant Hoses	Figure 65 Figure 66 Figure 67	2.74
Transmission Oil Cooler	Figure 68	6.49
Water Pump	NA	6.35
Coolant Overflow Chamber	Figure 69	2.00
Auxiliary Oil Cooler	Figure 70	6.50
Charge Air Cooler	Figure 71	14.18
Seals	NA	.01
Centrifugal Cooling Fan Subsystem	NA	26.41
Radiator Fan Bottom Shroud	Figure 72	1.78
Radiator Fan Top Shroud	Figure 73	1.76
Centrifugal Fan	Figure 74	11.56
Fan Clutch	Figure 75 (x4) Figure 76 Figure 77 Figure 78	11.30

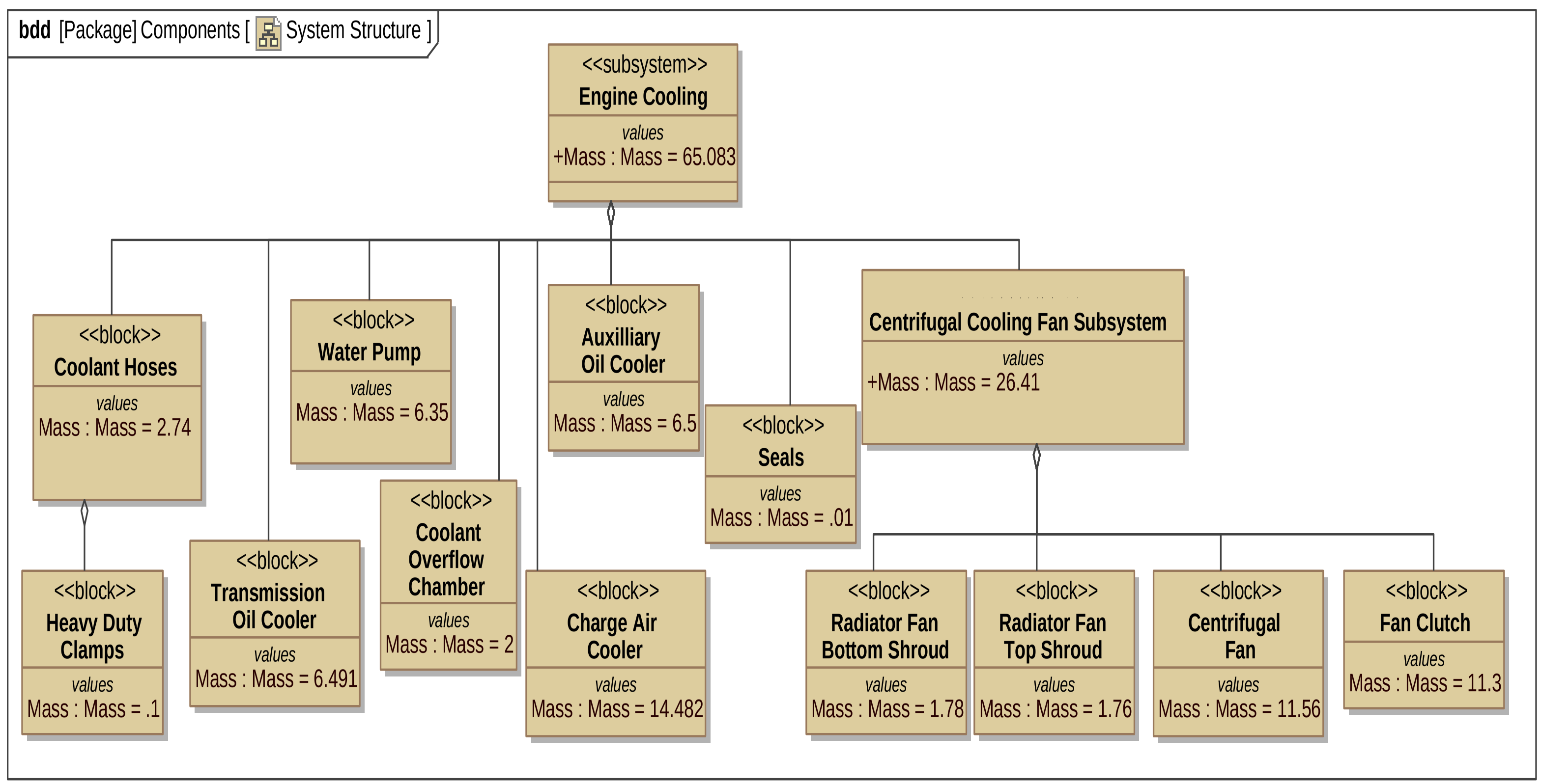


Figure 49: FMTV cooling subsystem component hierarchy

The component hierarchy is used in populating the RxC matrix when the subject of the requirement is something other than a leaf node of the hierarchy, like the engine cooling subsystem. A requirement with the subject as the engine cooling subsystem will be related to all components that are in that subsystem, all the leaf nodes that comes from that branch of the hierarchy. Once the components have been acquired, the requirements are related to the component hierarchy. An example of this is shown in Figure 50.

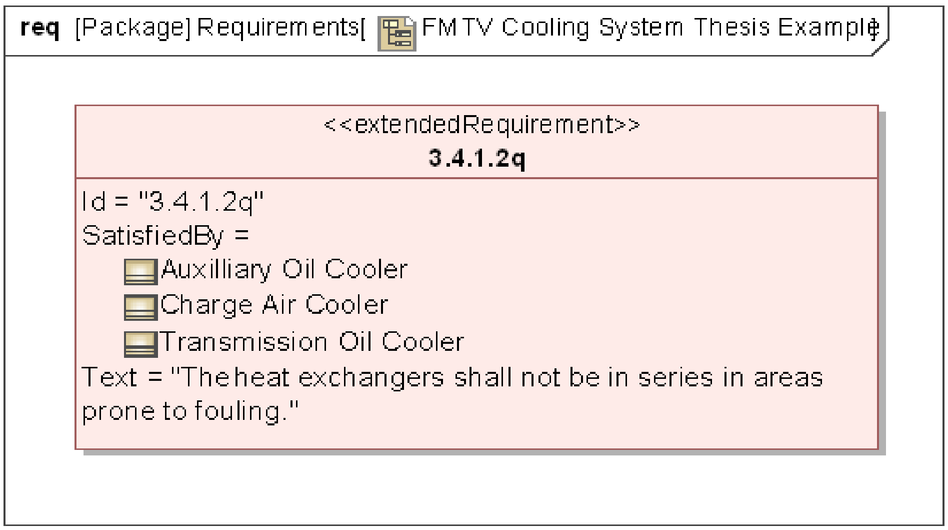
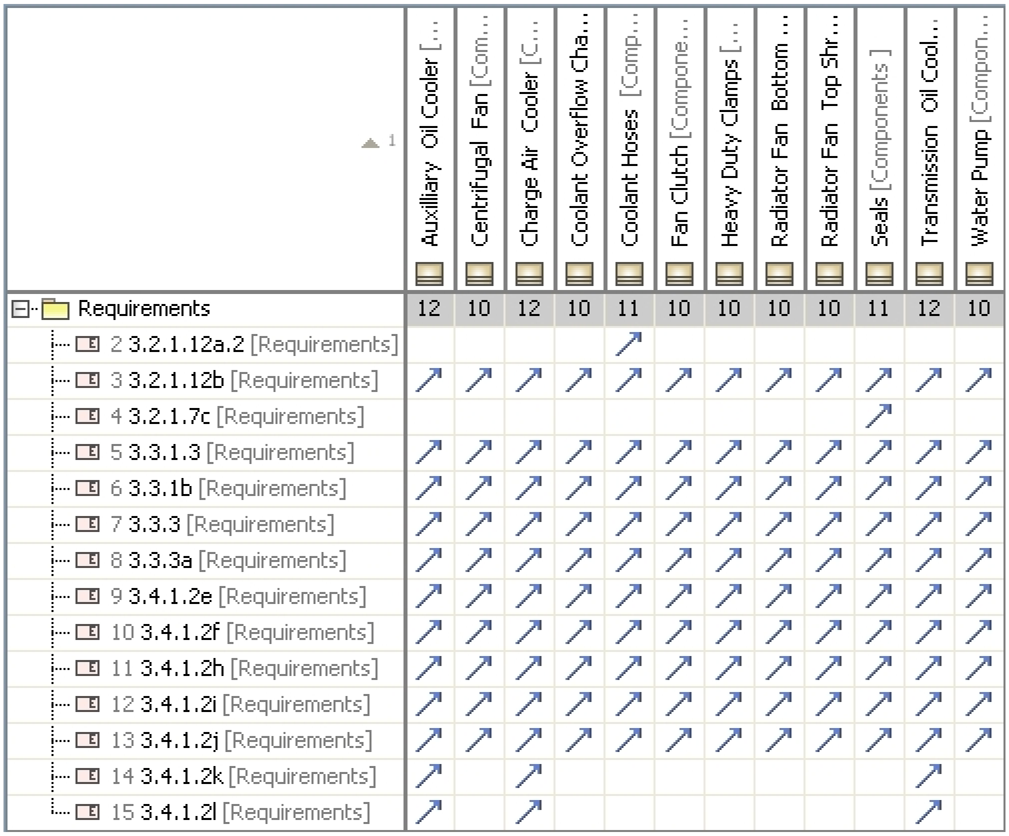


Figure 50: Example SysML requirements related to applicable components

Here, the requirement 3.4.1.22q (refer to Table 31 for definitions of the requirements) is related to the Auxiliary Oil Cooler, the Charge Air Cooler and the Transmission Oil Cooler. A DSM matrix of requirements and components is then constructed in the SysML software and is shown in Figure 51.



▲ 1	Auxilliary Oil Cooler [...	Centrifugal Fan [Com...	Charge Air Cooler [C...	Coolant Overflow Cha...	Coolant Hoses [Comp...	Fan Clutch [Compone...	Heavy Duty Clamps [...	Radiator Fan Bottom ...	Radiator Fan Top Shr...	Seals [Components]	Transmission Oil Cool...	Water Pump [Compon...
Requirements	12	10	12	10	11	10	10	10	10	11	12	10
2 3.2.1.12a.2 [Requirements]					↗							
3 3.2.1.12b [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
4 3.2.1.7c [Requirements]										↗		
5 3.3.1.3 [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
6 3.3.1b [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
7 3.3.3 [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
8 3.3.3a [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
9 3.4.1.2e [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
10 3.4.1.2f [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
11 3.4.1.2h [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
12 3.4.1.2i [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
13 3.4.1.2j [Requirements]	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
14 3.4.1.2k [Requirements]	↗		↗								↗	
15 3.4.1.2l [Requirements]	↗		↗								↗	

Figure 51: SysML DSM matrix of FMTV cooling subsystem

Once the SysML DSM has been created, it is exported to Excel to be used to construct the RxR matrices. The exported matrix in Excel represents relationships with non-numbers. These are changed to either binary or mass strengths. The two DSMs are shown in Table 33 and Table 34.

Table 33: FMTV cooling subsystem RxC binary matrix

	Auxiliary Oil Cooler	Centrifugal Fan	Charge Air Cooler	Coolant Overflow Chamber	Coolant Hoses	Fan Clutch	Heavy Duty Clamps	Radiator Fan Bottom Shroud	Radiator Fan Top Shroud	Seals	Transmission Oil Cooler	Water Pump
3.2.1.12a.2	0	0	0	0	1	0	0	0	0	0	0	0
3.2.1.12b	1	1	1	1	1	1	1	1	1	1	1	1
3.2.1.7c	0	0	0	0	0	0	0	0	0	1	0	0
3.3.1.3	1	1	1	1	1	1	1	1	1	1	1	1
3.3.1b	1	1	1	1	1	1	1	1	1	1	1	1
3.3.3	1	1	1	1	1	1	1	1	1	1	1	1
3.3.3a	1	1	1	1	1	1	1	1	1	1	1	1
3.4.1.2e	1	1	1	1	1	1	1	1	1	1	1	1
3.4.1.2f	1	1	1	1	1	1	1	1	1	1	1	1
3.4.1.2h	1	1	1	1	1	1	1	1	1	1	1	1
3.4.1.2i	1	1	1	1	1	1	1	1	1	1	1	1
3.4.1.2j	1	1	1	1	1	1	1	1	1	1	1	1
3.4.1.2k	1	0	1	0	0	0	0	0	0	0	1	0
3.4.1.2l	1	0	1	0	0	0	0	0	0	0	1	0
3.4.1.2m	0	0	0	0	0	0	1	0	0	0	0	0
3.4.1.2n	0	0	0	0	0	0	1	0	0	0	0	0
3.4.1.2o	0	0	0	0	0	0	1	0	0	0	0	0
3.4.1.2p	0	0	0	0	0	0	1	0	0	0	0	0
3.4.1.2q	1	0	1	0	0	0	0	0	0	0	1	0
3.4.1b	0	0	0	0	0	1	0	0	0	0	0	0
3.4.4b	0	0	0	0	0	0	0	0	0	0	1	0

Table 34: FMTV cooling subsystem RxC mass matrix

	Auxiliary Oil Cooler	Centrifugal Fan	Charge Air Cooler	Coolant Overflow Chamber	Coolant Hoses	Fan Clutch	Heavy Duty Clamps	Radiator Fan Bottom Shroud	Radiator Fan Top Shroud	Seals	Transmission Oil Cooler	Water Pump
Mass	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.2.1.12a.2	0.00	0.00	0.00	0.00	2.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3.2.1.12b	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.2.1.7c	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
3.3.1.3	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.3.1b	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.3.3	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.3.3a	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.4.1.2e	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.4.1.2f	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.4.1.2h	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.4.1.2i	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.4.1.2j	6.50	11.56	14.48	2.00	2.74	11.30	0.10	1.78	1.76	0.01	6.49	6.35
3.4.1.2k	6.50	0.00	14.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.49	0.00
3.4.1.2l	6.50	0.00	14.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.49	0.00
3.4.1.2m	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00
3.4.1.2n	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00
3.4.1.2o	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00
3.4.1.2p	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00
3.4.1.2q	6.50	0.00	14.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.49	0.00
3.4.1b	0.00	0.00	0.00	0.00	0.00	11.30	0.00	0.00	0.00	0.00	0.00	0.00
3.4.4b	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.49	0.00

The RxC binary matrix shows the existence of relationships while the RxC mass matrix shows a weighted relationship. To create the RxR mass matrix, the RxC mass matrix is multiplied with the transpose of the RxC binary matrix. For illustration purposes, the transpose is displayed in Table 35.

Table 35: FMTV cooling subsystem CxR binary matrix

	3.2.1.12a.2	3.2.1.12b	3.2.1.7c	3.3.1.3	3.3.1b	3.3.3	3.3.3a	3.4.1.2e	3.4.1.2f	3.4.1.2h	3.4.1.2i	3.4.1.2j	3.4.1.2k	3.4.1.2l	3.4.1.2m	3.4.1.2n	3.4.1.2o	3.4.1.2p	3.4.1.2q	3.4.1b	3.4.4b
Auxiliary Oil Cooler	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	0
Centrifugal Fan	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Charge Air Cooler	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	0
Coolant Overflow Chamber	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Coolant Hoses	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Fan Clutch	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0
Heavy Duty Clamps	0	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0
Radiator Fan Bottom Shroud	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Radiator Fan Top Shroud	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Seals	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Transmission Oil Cooler	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1
Water Pump	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0

The equation used to create the RxR mass matrix for the FMTV cooling subsystem is shown below.

$$R \times C\,Mass \;\times\; C \times R\,Mass \;=\; R \times R\,Mass \qquad \text{Equation 6.1}$$

The RxR mass matrix shows

1. The amount of mass each requirement affects (diagonal)
2. The amount of mass affected by two requirements (off-diagonal)

The resulting RxR mass matrix is shown in Table 36.

	3.2.1.12a.2	3.2.1.12b	3.2.1.7c	3.3.1.3	3.3.1b	3.3.3	3.3.3a	3.4.1.2e	3.4.1.2f	3.4.1.2h	3.4.1.2i	3.4.1.2j	3.4.1.2k	3.4.1.2l	3.4.1.2m	3.4.1.2n	3.4.1.2o	3.4.1.2p	3.4.1.2q	3.4.1b	3.4.4b
3.2.1.12a.2	3	3	0	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
3.2.1.12b	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.2.1.7c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.3.1.3	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.3.1b	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.3.3	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.3.3a	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2e	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2f	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2h	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2i	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2j	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2k	0	27	0	27	27	27	27	27	27	27	27	27	27	27	0	0	0	0	27	0	6
3.4.1.2l	0	27	0	27	27	27	27	27	27	27	27	27	27	27	0	0	0	0	27	0	6
3.4.1.2m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4.1.2n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4.1.2o	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4.1.2p	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4.1.2q	0	27	0	27	27	27	27	27	27	27	27	27	27	27	0	0	0	0	27	0	6
3.4.1b	0	11	0	11	11	11	11	11	11	11	11	11	0	0	0	0	0	0	0	11	0
3.4.4b	0	6	0	6	6	6	6	6	6	6	6	6	6	6	0	0	0	0	6	0	6

Table 36: FMTV cooling subsystem RxR mass matrix

The RxR binary matrix was created by multiplying the RxC binary matrix with the CxR binary matrix. This matrix shows

3. The number of components each requirement affects (diagonal)
4. The number of components affected by two requirements (off-diagonal)

The FMTV cooling subsystem RxR binary matrix is shown in Table 37.

Table 37: FMTV cooling subsystem RxR binary matrix

	3.2.1.12a.2	3.2.1.12b	3.2.1.7c	3.3.1.3	3.3.1b	3.3.3	3.3.3a	3.4.1.2e	3.4.1.2f	3.4.1.2h	3.4.1.2i	3.4.1.2j	3.4.1.2k	3.4.1.2l	3.4.1.2m	3.4.1.2n	3.4.1.2o	3.4.1.2p	3.4.1.2q	3.4.1b	3.4.4b
3.2.1.12a.2	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
3.2.1.12b	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.2.1.7c	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
3.3.1.3	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.3.1b	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.3.3	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.3.3a	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.4.1.2e	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.4.1.2f	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.4.1.2h	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.4.1.2i	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.4.1.2j	1	12	1	12	12	12	12	12	12	12	12	12	3	3	1	1	1	1	3	1	1
3.4.1.2k	0	3	0	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	3	0	1
3.4.1.2l	0	3	0	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	3	0	1
3.4.1.2m	0	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0
3.4.1.2n	0	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0
3.4.1.2o	0	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0
3.4.1.2p	0	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0
3.4.1.2q	0	3	0	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	3	0	1
3.4.1b	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0
3.4.4b	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1

Step 3: Requirement Analysis and Identification of Mass Intensive Requirements

Using the RxR binary and mass matrices, the total mass affected and the coupling levels were attained and compiled in Table 38.

Table 38: Requirement mass and coupling data for FMTV cooling subsystem

	total mass affected by 1 requirement	total mass coupled to (w/other requirements)	# requirements related to
3.2.1.12a.2	2.74	27.40	11
3.2.1.12b	65.07	688.98	21
3.2.1.7c	0.01	0.10	11
3.3.1.3	65.07	688.98	21
3.3.1b	65.07	688.98	21
3.3.3	65.07	688.98	21
3.3.3a	65.07	688.98	21
3.4.1.2e	65.07	688.98	21
3.4.1.2f	65.07	688.98	21
3.4.1.2h	65.07	688.98	21
3.4.1.2i	65.07	688.98	21
3.4.1.2j	65.07	688.98	21
3.4.1.2k	27.47	336.13	14
3.4.1.2l	27.47	336.13	14
3.4.1.2m	0.10	1.30	14
3.4.1.2n	0.10	1.30	14
3.4.1.2o	0.10	1.30	14
3.4.1.2p	0.10	1.30	14
3.4.1.2q	27.47	336.13	14
3.4.1b	11.3	113.00	11
3.4.4b	6.49	84.37	14

The values in the column showing "total mass affected by 1 requirement" are calculated by taking the diagonal values for each requirement from the RxR mass matrix in Table 36. The values in the column showing "total mass coupled to (w/ other requirements)" are calculated by taking the sum of the rows in the RxR mass matrix in Table 36 and subtracting the value on the diagonal.

This shows the amount of mass the selected requirement and all other requirements coupled to it affect. The values in the column showing "# requirements related to" shows the coupling of each requirement. They are found by counting the number of nonzero values in each row of the RxR binary matrix.

Taking into account mass and coupling leads to several different combination possibilities: requirements that affect much mass and are also highly coupled, requirements that affect much mass and are lowly coupled, requirements that affect little mass and are highly coupled and requirements that affect little mass and are lowly coupled. Also to be included are mid-level mass or coupling values. These are always treated as second-choice options if the best case option is not available. This is graphically shown in Figure 52.

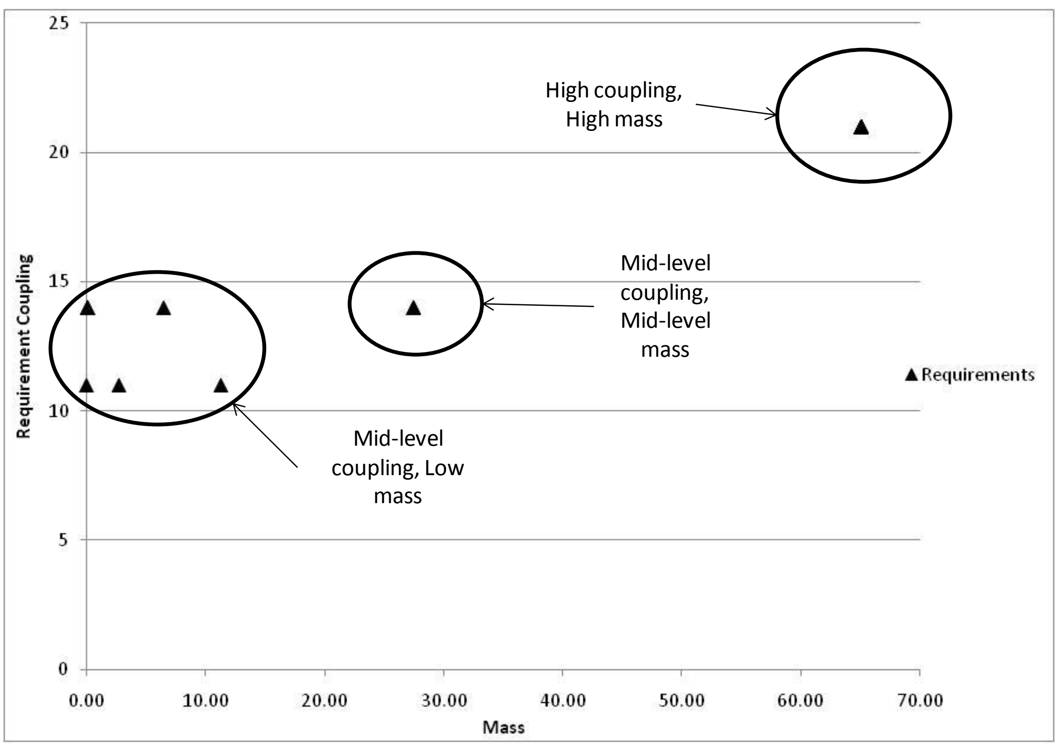


Figure 52: Requirement coupling vs. mass for FMTV engine cooling subsystem

This figure shows the cooling requirements fit into three categories: high mass high coupling, mid-level coupling mid-level mass and mid-level coupling low mass. Each data point stands for the group of requirements that have the same values for requirement mass and coupling. For example, the high coupling high mass data point represents requirements 3.3.1.3, 3.3.1b, 3.3.3, 3.3.3a, 3.4.1.2e, 3.4.1.2f, 3.4.1.2h, 3.4.1.2i, 3.4.1.2j and 3.2.1.12b from the requirements list which both have mass values of 65 and coupling values of 21.

Given this discussion of how requirements are selected to change in Chapter 3, the following Table 39 shows the order in which requirements should be modified to most efficiently reduce mass.

Table 39: Order priority for which requirements to change of FMTV cooling subsystem

	total mass affected by 1 requirement	total mass coupled to (w/other requirements)	# requirements related to
3.2.1.12b	65.07	688.98	21
3.3.1.3	65.07	688.98	21
3.3.1b	65.07	688.98	21
3.3.3	65.07	688.98	21
3.3.3a	65.07	688.98	21
3.4.1.2e	65.07	688.98	21
3.4.1.2f	65.07	688.98	21
3.4.1.2h	65.07	688.98	21
3.4.1.2i	65.07	688.98	21
3.4.1.2j	65.07	688.98	21
3.4.1.2k	27.47	336.13	14
3.4.1.2l	27.47	336.13	14
3.4.1.2q	27.47	336.13	14
3.4.1b	11.3	113.00	11
3.4.4b	6.49	84.37	14
3.2.1.12a.2	2.74	27.40	11
3.4.1.2m	0.10	1.30	14
3.4.1.2n	0.10	1.30	14
3.4.1.2o	0.10	1.30	14
3.4.1.2p	0.10	1.30	14
3.2.1.7c	0.01	0.10	11

Requirements that affect high mass and are highly coupled are changed first while requirements that affect little mass and are lowly coupled are changed last.

This chapter has shown an example of an FMTV subsystem to prove the usefulness of this proposed requirement method to identify requirements that affect significant amounts of mass. The next chapter will show another FMTV subsystem to prove the method is useful for other design subsystems also.

Chapter 7. Analysis of FMTV Chassis Subsystem

The requirements modeling method developed in Chapter 3 is used to analyze the FMTV Chassis Subsystem. It is displayed for review purposes in Figure 53.

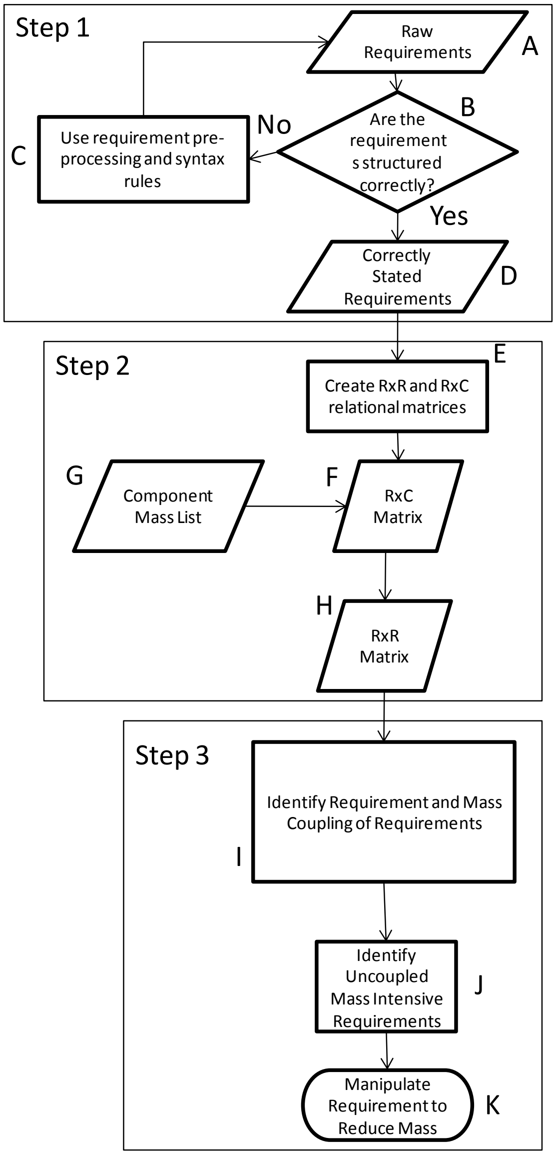


Figure 53: Requirement analysis flowchart

In this chapter the FMTV chassis subsystem will be discussed. No chassis subsystem view was available, but pictures of the components are included in Appendix 1.

The data for the analysis is obtained from several different sources including (1) ATPD2131F.1 and (2) CAD models. The mass reduction method for each subsystem is described step by step in the following sections.

Step 1: Acquire and process requirements

The unprocessed FMTV Cooling System Requirements used in the requirements analysis method are included in Table 40.

Table 40: Unprocessed FMTV chassis subsystem requirements

No.	Text
3.2.1.12	Any components exposed up to XX inches from the ground with the emergency CTIS setting in force, to include hoses, cables, lanyards, lines, tanks, valves, wires, cylinders, boxes, shall be shielded or able to withstand, going in forward or reverse, with no degradation of vehicle operation: the repeated impact of brush and tree branches; dry debris raised by cross country operation; soil scraping at XX mph.
3.3.1	All materials shall be new and unused.
3.3.3	Workmanship shall be of the highest grade consistent with the intention of this specification. Each vehicle shall have no evidence of cracks, dents, scratches, burrs, sharp edges, loose parts, foreign matter, or any other evidence of poor workmanship that shall render the vehicle unsuitable/unsafe for the purpose intended.
3.4.1	The fan clutch shall be such that, in the event of failure, the fan shall be constantly engaged.
3.4.1.2	The cooling system shall be capable of retention and recovery of XX% coolant overflow or have XX% expansion reserve capacity. The cooling system shall be capable of continuous de-aeration of XX cfm of air per cylinder at rated engine speed at any slope the vehicle is required to operate on. The system shall fill completely, with an automatic de-aeration feature to preclude air cavitation at any coolant fill rate up to the maximum fill rate. Maintain the specified component operating temperatures within the specified limits while operating continuously at full load and XX tractive effort to gross vehicle weight ratio (TE/GVW) while under the maximum conditions of XXo F for all models with the exception of the Expansible Van, LHS, Tractor and Wrecker which shall meet a minimum of XX TE/GVW while under maximum conditions of XXo F. Does not exceed temperature limits while operating at rated engine power. Meets the requirements after a drawdown of XX% of engine coolant. Specified fluid temperatures shall not exceed the lower of those for which the component manufacturer shall provide warranty, or the following: REFER TO PG 21. The radiator shall have a maximum of XX fins per cm and shall be located to minimize air side fouling. Heavy duty clamps shall be used, shall be clearly visible, located for ease of connection, and ensure positive sealing. The cooling system shall not be comprised of heat exchangers in series in areas prone to fouling.

After applying the pre-processing and syntax rules the following structured requirements in Table 41 were written.

Table 41: Structured FMTV chassis system requirements

No.	Text
3.2.1.12a	Components shall be shielded or able to withstand, going forward, with no degradation of vehicle operation: the repeated impact of brush and tree branches; dry debris raised by cross country operation; soil scraping at XX mph while exposed up to XX inches from the ground with the emergency CTIS setting in force, to include hoses, cables, lanyards, lines, tanks, valves, wires, cylinders and boxes.
3.2.1.12b	Components shall be shielded or able to withstand, going in reverse, with no degradation of vehicle operation: the repeated impact of brush and tree branches; dry debris raised by cross country operation; soil scraping at XX mph while exposed up to XX inches from the ground with the emergency CTIS setting in force to include hoses, cables, lanyards, lines, tanks, valves, wires, cylinders and boxes .
3.2.1.12.1	The basic chassis shall function when exposed to emissions from Electromagnetic Compatibility (EMC) and Near Strike Lightning (NSL).
3.2.1.15a	The vehicle shall have an approach angle a minimum of XX° for all models with kits and winches.
3.2.1.15b	The vehicle shall have an approach angle a minimum of XX° for all models with kits and without winches.
3.2.1.15c	The vehicle shall have an approach angle a minimum of XX° for all models without kits and with winches.
3.2.1.15d	The vehicle shall have an approach angle a minimum of XX° for all models without kits and without winches.
3.2.1.15e	The vehicle shall have a departure angle a minimum of XX° for all basic cargo trucks with kits and winches .
3.2.1.15f	The vehicle shall have a departure angle a minimum of XX° for basic cargo trucks with kits and without winches.
3.2.1.15g	The vehicle shall have a departure angle a minimum of XX° for basic cargo trucks without kits and with winches.
3.2.1.15h	The vehicle shall have a departure angle a minimum of XX° for all basic cargo trucks without kits and without winches.
3.2.1.16a	The vehicle shall have a minimum ground clearance between front and rear tires of not less than XX inches (XX cm), with kits, with tire pressures at highway mode.
3.2.1.16b	The vehicle shall have a minimum ground clearance between front and rear tires of not less than XX inches (XX cm), without kits, with tire pressures at highway mode.
3.2.1.17	The vehicle shall tow a like vehicle (see paragraph 6.3.14) at GVW for a distance of at least XX miles at a speed of XX mph, without preparation, without degradation or damage to either vehicle.
3.2.2.3.3a	The vehicle frame shall resist corrosion.
3.2.2.3.3b	The vehicle sub-framing shall resist corrosion.
3.2.2.3.5	Dissimilar metals shall be electrically isolated to prevent galvanic corrosion.
3.2.8.3	The vehicle shall have a maximum height less than XX in. (XX cm) for AD models.
3.3.1a	Radioactive materials shall not be used.
3.3.1b	All component materials shall be new and unused.
3.4.4a	The frames shall employ structural members which provide optimum section efficiency for torsional stiffness.

3.4.4b	The frames shall employ structural members which provide optimum section efficiency for bending stiffness.
3.4.4c	Frame shall prevent permanent torsional warping due to bending throughout the operating profile of the vehicle (see Table III-IX).
3.4.4d	Frame shall prevent permanent torsional twist due to bending throughout the operating profile of the vehicle (see Table III-IX).
3.4.4e	Frame shall prevent permanent deflection due to bending throughout the operating profile of the vehicle (see Table III-IX).
3.4.5.1	The suspension design shall limit the vertical natural frequency of the sprung mass to a maximum of XX hertz.

These chassis subsystem requirements refer not only to the leaf node parts of the physical design (components) but to the branch as well (chassis subsystem).

Once the requirements have been standardized, the next step of obtaining component information is started.

Step 2: Map Requirements to Components

Once the Requirement Analysis Method was developed, it was implemented on the FMTV chassis subsystems. To implement the requirement analysis method on the FMTV chassis subsystems, the component masses were acquired and are shown in Table 42.

Table 42: FMTV chassis subsystem component list

Component	Figure	Mass (kg)
Trailer Hitch	Figure 79	11.90
Rear Axle	NA	317.51
Rear Axle Housing	Figure 80	136.72
Front Axle	NA	317.51
Front Axle Housing	Figure 81	136.72
Frame	Figure 82	431.78
Leaf Springs	Figure 83	224.86
Fifth Wheel	Figure 84	70.75
Tires	Figure 85	144.70x4
Winch	NA	35.00

Some of the subsystem components were not found to have a .prt file and thus comparable parts were found on the internet and their masses were used. These components are marked with an NA for not applicable since an image was not created in SolidWorks of that file.

These chassis subsystem requirements refer not only to leaf node parts of the physical design (components), but to the branch as well (chassis subsystem). Hence, a hierarchy is represented in the requirement list. By Rule 1 of the requirement preprocessing rules, the subject of the requirement has to be part of the physical subsystem. This physical subsystem is a hierarchy also and is modeled in MagicDraw SysML and is included in Figure 54.

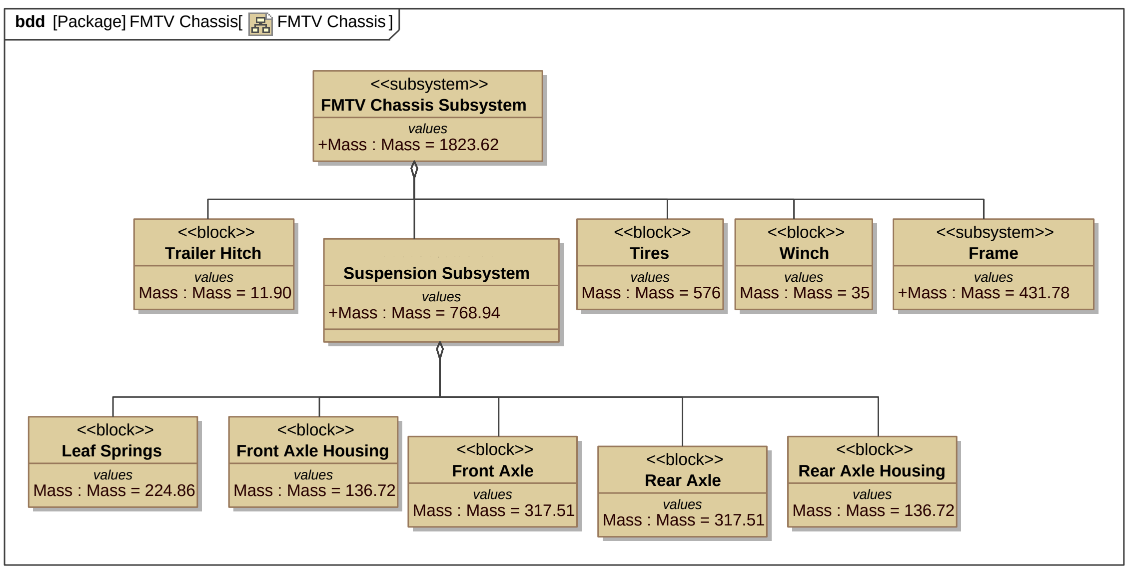


Figure 54: FMTV chassis subsystem

The component hierarchy is used in populating the RxC matrix when the subject of the requirement is something other than a leaf node of the hierarchy, like the FMTV chassis subsystem. A requirement with the subject as the FMTV chassis subsystem will be related to all components that are in that subsystem, all the leaf nodes that comes from that branch of the hierarchy. Once the components have been acquired, the requirements are related to the component hierarchy. An example of a chassis requirement related to the component hierarchy is shown in Figure 55.

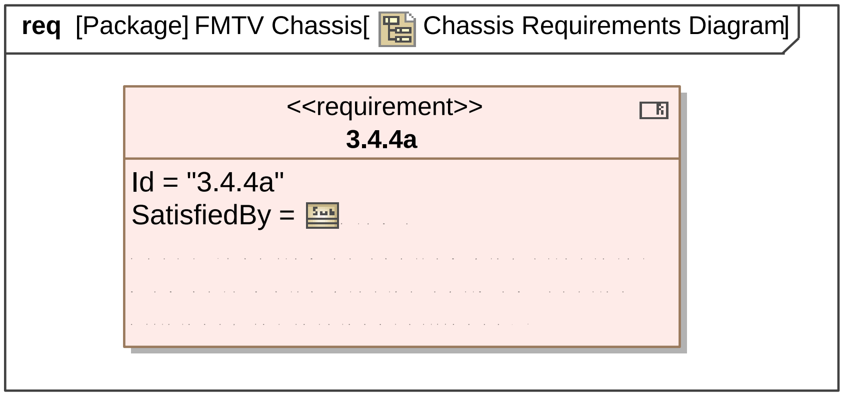


Figure 55: Example SysML FMTV chassis subsystem requirement with component relationships

Here, the requirement 3.4.4a (refer to Table 41 for definitions of the requirements) is related to the Frame. A DSM matrix of requirements and components is then constructed in the SysML software and is shown in Figure 51.

	Fifth Wheel [Com…	Front Axle [Comp…	Front Axle Housin…	Leaf Springs [Co…	Main Beams [Com…	Rear Axle [Comp…	Rear Axle Housin…	Tires [Component…	Trailer Hitch [Com…	Winch [Compone…
Chassis [Requirements]	15	16	16	16	21	16	16	15	15	15
32 3.2.1.12.1 [Requiremen…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
29 3.2.1.12a [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
30 3.2.1.12c [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
31 3.2.1.12d [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
33 3.2.1.15a [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
34 3.2.1.15b [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
35 3.2.1.15c [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
36 3.2.1.15d [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
37 3.2.1.15e [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
38 3.2.1.15f [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
39 3.2.1.15g [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
40 3.2.1.15h [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
41 3.2.1.16a [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
42 3.2.1.16b [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
43 3.2.1.17 [Requirements…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
44 3.2.2.3.3a [Requiremen…					↗					
45 3.2.2.3.3b [Requiremen…					↗					
74 3.4.4a [Requirements::…					↗					
49 3.4.4c [Requirements::…					↗					
50 3.4.4d [Requirements::…					↗					
51 3.4.4e [Requirements::…					↗					
52 3.4.5.1 [Requirements::…		↗	↗	↗		↗	↗			
Shared [Requirements]	5	5	5	5	6	5	5	5	5	5
3 3.2.1.12b [Requirements…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
46 3.2.2.3.5 [Requirement…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
47 3.2.8.3 [Requirements::…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
48 3.3.1a [Requirements::…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
6 3.3.1b [Requirements::S…	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗
22 3.4.4b [Requirements::…					↗					

Figure 56: SysML DSM matrix of FMTV chassis subsystem

Once the SysML DSM has been created, it is exported to Excel to be used to construct the RxR matrices. The exported matrix in Excel represents relationships with non-numbers. These are changed to either binary or mass strengths. The two DSMs are shown in Table 43 and

Table 44.

Table 43 : FMTV chassis subsystem RxC binary matrix

	trailer hitch	rear axle	rear axle housing	front axle	front axle housing	main beams subsystem	leaf springs	fifth wheel	tires	winch
3.2.1.12a	1	1	1	1	1	1	1	1	1	1
3.2.1.12b	1	1	1	1	1	1	1	1	1	1
3.2.1.12c	1	1	1	1	1	1	1	1	1	1
3.2.1.12d	1	1	1	1	1	1	1	1	1	1
3.2.1.12.1	1	1	1	1	1	1	1	1	1	1
3.2.1.15a	1	1	1	1	1	1	1	1	1	1
3.2.1.15b	1	1	1	1	1	1	1	1	1	1
3.2.1.15c	1	1	1	1	1	1	1	1	1	1
3.2.1.15d	1	1	1	1	1	1	1	1	1	1
3.2.1.15e	1	1	1	1	1	1	1	1	1	1
3.2.1.15f	1	1	1	1	1	1	1	1	1	1
3.2.1.15g	1	1	1	1	1	1	1	1	1	1
3.2.1.15h	1	1	1	1	1	1	1	1	1	1
3.2.1.16a	1	1	1	1	1	1	1	1	1	1
3.2.1.16b	1	1	1	1	1	1	1	1	1	1
3.2.1.17	1	1	1	1	1	1	1	1	1	1
3.2.2.3.3a	0	0	0	0	0	1	0	0	0	0
3.2.2.3.3b	0	0	0	0	0	1	0	0	0	0
3.2.2.3.5	1	1	1	1	1	1	1	1	1	1
3.2.8.3	1	1	1	1	1	1	1	1	1	1
3.3.1a	1	1	1	1	1	1	1	1	1	1
3.3.1b	1	1	1	1	1	1	1	1	1	1
3.4.4a	0	0	0	0	0	1	0	0	0	0
3.4.4b	0	0	0	0	0	1	0	0	0	0
3.4.4c	0	0	0	0	0	1	0	0	0	0
3.4.4d	0	0	0	0	0	1	0	0	0	0
3.4.4e	0	0	0	0	0	1	0	0	0	0
3.4.5.1	0	1	1	1	1	0	1	0	0	0

Table 44: FMTV chassis subsystem RxC mass matrix

	trailer hitch	rear axle	rear axle housing	front axle	front axle housing	main beams subsystem	leaf springs	fifth wheel	tires	winch
Mass	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.12a	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.12b	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.12c	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.12d	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.12.1	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.15a	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.15b	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.15c	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.15d	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.15e	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.15f	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.15g	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.15h	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.16a	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.16b	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.1.17	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.2.3.3a	0.00	0.00	0.00	0.00	0.00	431.78	0.00	0.00	0.00	0.00
3.2.2.3.3b	0.00	0.00	0.00	0.00	0.00	431.78	0.00	0.00	0.00	0.00
3.2.2.3.5	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.2.8.3	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.3.1a	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.3.1b	11.90	135.32	136.72	135.32	136.72	431.78	224.86	70.75	144.70	35.00
3.4.4a	0.00	0.00	0.00	0.00	0.00	431.78	0.00	0.00	0.00	0.00
3.4.4b	0.00	0.00	0.00	0.00	0.00	431.78	0.00	0.00	0.00	0.00
3.4.4c	0.00	0.00	0.00	0.00	0.00	431.78	0.00	0.00	0.00	0.00
3.4.4d	0.00	0.00	0.00	0.00	0.00	431.78	0.00	0.00	0.00	0.00
3.4.4e	0.00	0.00	0.00	0.00	0.00	431.78	0.00	0.00	0.00	0.00
3.4.5.1	0.00	135.32	136.72	135.32	136.72	0.00	224.86	0.00	0.00	0.00

The RxC binary matrix shows the existence of relationships while the RxC mass matrix shows a weighted relationship. To create the RxR mass matrix, the RxC mass matrix is multiplied with the transpose of the RxC binary matrix. For illustration purposes, the transpose is displayed in Table 45.

	3.2.1.12a.2	3.2.1.12b	3.2.1.7c	3.3.1.3	3.3.1b	3.3.3	3.3.3a	3.4.1.2e	3.4.1.2f	3.4.1.2h	3.4.1.2i	3.4.1.2j	3.4.1.2k	3.4.1.2l	3.4.1.2m	3.4.1.2n	3.4.1.2o	3.4.1.2p	3.4.1.2q	3.4.1b	3.4.4b
3.2.1.12a.2	3	3	0	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
3.2.1.12b	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.2.1.7c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.3.1.3	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.3.1b	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.3.3	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.3.3a	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2e	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2f	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2h	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2i	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2j	3	65	0	65	65	65	65	65	65	65	65	65	27	27	0	0	0	0	27	11	6
3.4.1.2k	0	27	0	27	27	27	27	27	27	27	27	27	27	27	0	0	0	0	27	0	6
3.4.1.2l	0	27	0	27	27	27	27	27	27	27	27	27	27	27	0	0	0	0	27	0	6
3.4.1.2m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4.1.2n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4.1.2o	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4.1.2p	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4.1.2q	0	27	0	27	27	27	27	27	27	27	27	27	27	27	0	0	0	0	27	0	6
3.4.1b	0	11	0	11	11	11	11	11	11	11	11	11	0	0	0	0	0	0	0	11	0
3.4.4b	0	6	0	6	6	6	6	6	6	6	6	6	6	6	0	0	0	0	6	0	6

Table 45: FMTV chassis subsystem CxR matrix

The equation used to create the RxR mass matrix for the FMTV cooling subsystem is shown below.

$$R \times C\,Mass \;\times\; C \times R\,Mass \;=\; R \times R\,Mass \qquad \text{Equation 7.1}$$

The RxR mass matrix shows

1. The amount of mass each requirement affects (diagonal)
2. The amount of mass affected by two requirements (off-diagonal)

The resulting RxR mass matrix is shown in Table 46. Due to the very large size of the table, only an excerpt is shown.

Table 46: Excerpt of FMTV chassis subsystem RxR mass matrix

	3.2.1.12a	3.2.1.12b	3.2.1.12c	3.2.1.12d	3.2.1.12.1
3.2.1.12a	1463	1463	1463	1463	1463
3.2.1.12b	1463	1463	1463	1463	1463
3.2.1.12c	1463	1463	1463	1463	1463
3.2.1.12d	1463	1463	1463	1463	1463
3.2.1.12.1	1463	1463	1463	1463	1463
3.2.1.15a	1463	1463	1463	1463	1463
3.2.1.15b	1463	1463	1463	1463	1463
3.2.1.15c	1463	1463	1463	1463	1463
3.2.1.15d	1463	1463	1463	1463	1463
3.2.1.15e	1463	1463	1463	1463	1463
3.2.1.15f	1463	1463	1463	1463	1463
3.2.1.15g	1463	1463	1463	1463	1463
3.2.1.15h	1463	1463	1463	1463	1463
3.2.1.16a	1463	1463	1463	1463	1463
3.2.1.16b	1463	1463	1463	1463	1463
3.2.1.17	1463	1463	1463	1463	1463
3.2.2.3.3a	432	432	432	432	432
3.2.2.3.3b	432	432	432	432	432

The RxR binary matrix was created by multiplying the RxC binary matrix with the CxR binary matrix. This matrix shows

1. The number of components each requirement affects (diagonal)
2. The number of components affected by two requirements (off-diagonal)

The FMTV cooling subsystem RxR binary matrix is shown in Table 47.

Table 47: FMTV chassis subsytem RxR binary matrix

	3.2.1.12a	3.2.1.12b	3.2.1.12c	3.2.1.12d	3.2.1.12.1	3.2.1.15a	3.2.1.15b	3.2.1.15c	3.2.1.15d	3.2.1.15e	3.2.1.15f	3.2.1.15g	3.2.1.15h	3.2.1.16a	3.2.1.16b	3.2.1.17	3.2.2.3.3a	3.2.2.3.3b	3.2.2.3.5	3.2.8.3	3.3.1a	3.3.1b	3.4.4a	3.4.4b	3.4.4c	3.4.4d	3.4.4e	3.4.5.1
3.2.1.12a	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.12b	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.12c	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.12d	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.12.1	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.15a	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.15b	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.15c	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.15d	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.15e	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.15f	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.15g	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.15h	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.16a	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.16b	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.1.17	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.2.3.3a	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
3.2.2.3.3b	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
3.2.2.3.5	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.2.8.3	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.3.1a	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.3.1b	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	10	10	1	2	1	1	1	5
3.4.4a	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
3.4.4b	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
3.4.4c	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
3.4.4d	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
3.4.4e	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
3.4.5.1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	0	0	5	5	5	5	0	1	0	0	0	5

Step 3: Requirement Analysis and Identification of Mass Intensive Requirements

Using the RxR binary and mass matrices, the total mass affected and the coupling levels were attained and compiled in Table 48.

Table 48: Requirement mass and coupling data for FMTV chassis subsystem

	total mass affected by 1 requirement	total mass coupled to (w/other requirements)	# requirements related to
3.2.1.12a	1463	31815	28
3.2.1.12b	1463	31815	28
3.2.1.12c	1463	31815	28
3.2.1.12d	1463	31815	28
3.2.1.12.1	1463	31815	28
3.2.1.15a	1463	31815	28
3.2.1.15b	1463	31815	28
3.2.1.15c	1463	31815	28
3.2.1.15d	1463	31815	28
3.2.1.15e	1463	31815	28
3.2.1.15f	1463	31815	28
3.2.1.15g	1463	31815	28
3.2.1.15h	1463	31815	28
3.2.1.16a	1463	31815	28
3.2.1.16b	1463	31815	28
3.2.1.17	1463	31815	28
3.2.2.3.3a	432	11226	27
3.2.2.3.3b	432	11226	27
3.2.2.3.5	1463	31815	28
3.2.8.3	1463	31815	28
3.3.1a	1463	31815	28
3.3.1b	1463	31815	28
3.4.4a	432	11226	27
3.4.4b	432	11226	27
3.4.4c	432	11226	27
3.4.4d	432	11226	27
3.4.4e	432	11226	27
3.4.5.1	769	15604	22

The values in the column showing "total mass affected by 1 requirement" are calculated by taking the diagonal values for each requirement from the RxR mass matrix in Table 46. The values in the column showing “total mass coupled to (w/ other requirements)” are calculated by taking the sum of the rows in the RxR mass matrix in Table 46 and subtracting the value on the diagonal. This shows the amount of mass the selected requirement and all other requirements coupled to it affect. The values in the column showing “# requirements related to” shows the coupling of each requirement. They are found by counting the number of nonzero values in each row of the RxR binary matrix.

Taking into account mass and coupling leads to several different combination possibilities: requirements that affect much mass and are also highly coupled, requirements that affect much mass and are lowly coupled, requirements that affect little mass and are highly coupled and requirements that affect little mass and are lowly coupled. Also to be included are mid-level mass or coupling values. These are always treated as second-choice options if the best case option is not available. This is graphically shown in Figure 57.

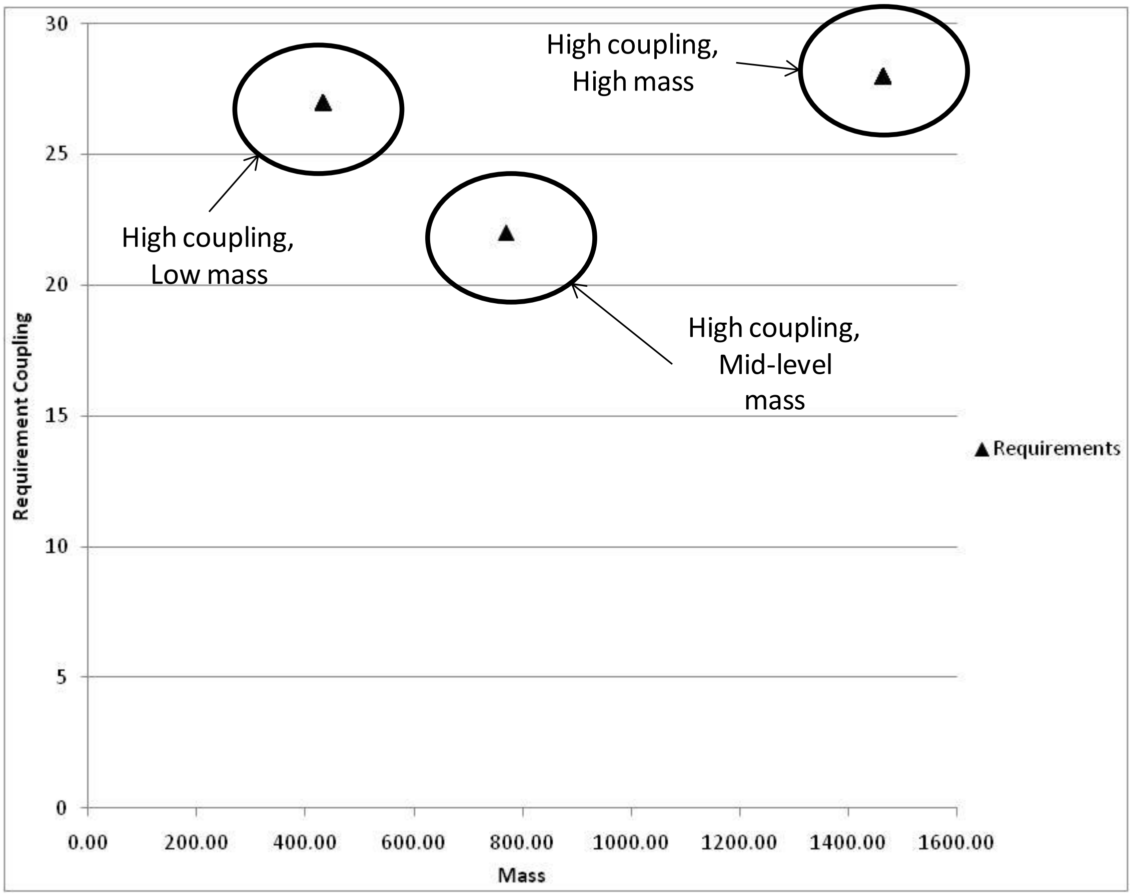


Figure 57: Requirement coupling vs. mass for FMTV chassis subsystem

This figure shows the chassis requirements fit into three categories: high mass high coupling, high coupling mid-level mass and high coupling low mass. Each data point stands for the group of requirements that have the same values for requirement mass and coupling. For example, the high coupling high mass data point represents requirements 3.2.1.12a, 3.2.1.12b, 3.2.1.12c, 3.2.1.12d, 3.2.1.12.1, 3.2.1.15a, 3.2.1.15b, 3.2.1.15c, 3.2.1.15d, 3.2.1.15e, 3.2.1.15f, 3.2.1.15g, 3.2.1.15h, 3.2.1.16a, 3.2.1.16b, 3.2.1.17, 3.2.2.3.5, 3.2.8.3, 3.3.1a and 3.3.1b from the requirements list which both have mass values of 1463 and coupling values of 28.

Given this discussion of how requirements are selected to change in Chapter 3, the following Table 49 shows the order in which requirements should be modified to most efficiently reduce mass

Table 49: Order priority for which requirements to change of FMTV chassis subsystem

	total mass affected by 1 requirement	total mass coupled to (w/other requirements)	# requirements related to
3.2.1.12a	1463.08	31814.66	28
3.2.1.12b	1463.08	31814.66	28
3.2.1.12c	1463.08	31814.66	28
3.2.1.12d	1463.08	31814.66	28
3.2.1.12.1	1463.08	31814.66	28
3.2.1.15a	1463.08	31814.66	28
3.2.1.15b	1463.08	31814.66	28
3.2.1.15c	1463.08	31814.66	28
3.2.1.15d	1463.08	31814.66	28
3.2.1.15e	1463.08	31814.66	28
3.2.1.15f	1463.08	31814.66	28
3.2.1.15g	1463.08	31814.66	28
3.2.1.15h	1463.08	31814.66	28
3.2.1.16a	1463.08	31814.66	28
3.2.1.16b	1463.08	31814.66	28
3.2.1.17	1463.08	31814.66	28
3.2.2.3.5	1463.08	31814.66	28
3.2.8.3	1463.08	31814.66	28
3.3.1a	1463.08	31814.66	28
3.3.1b	1463.08	31814.66	28
3.4.5.1	768.95	15603.80	22
3.4.4a	431.78	11226.15	27
3.4.4b	431.78	11226.15	27
3.4.4c	431.78	11226.15	27
3.4.4d	431.78	11226.15	27
3.4.4e	431.78	11226.15	27
3.2.2.3.3a	431.78	11226.15	27
3.2.2.3.3b	431.78	11226.15	27

Requirements that affect high mass and are highly coupled are changed first while requirements that affect little mass and are lowly coupled are changed last.

This chapter has shown an example of an FMTV subsystem to prove the usefulness of this proposed requirement method to identify requirements that affect significant amounts of mass. The next chapter will show another FMTV subsystem to prove the method is useful for other design subsystems also.

Chapter 8. Analysis of FMTV Cab Subsystem

In this chapter the FMTV Cab Subsystem will be discussed. The data for the analysis is obtained from several different sources including (1) ATPD2131F.1 and (2) CAD models. For reference purposes, the requirement analysis method is again listed below in Figure 58.

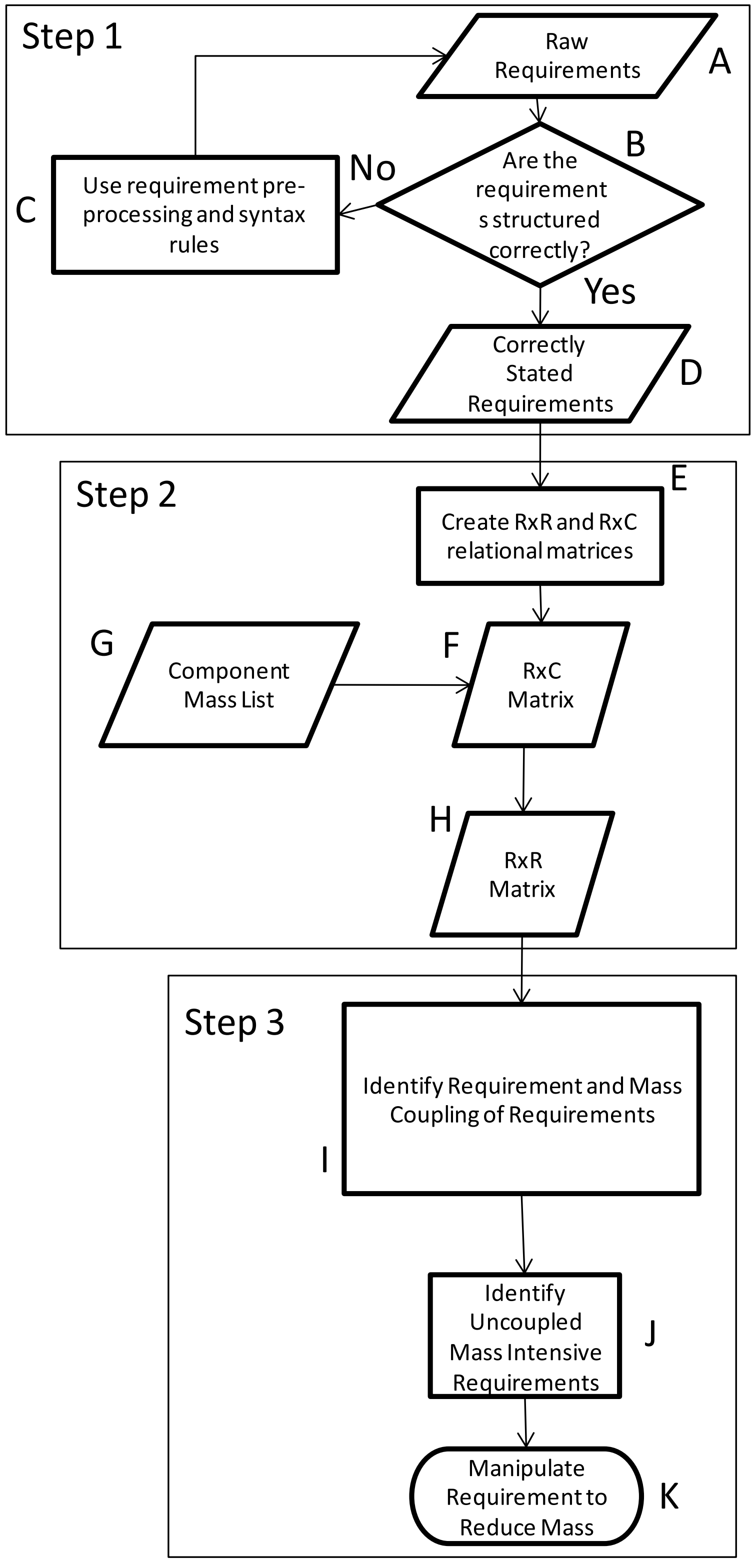


Figure 58: Requirement analysis flowchart

Step 1: Acquire and process requirements

The unprocessed FMTV Cab Subsystem Requirements used in the requirements analysis method are included in Table 50.

Table 50: Unprocessed FMTV chassis subsystem requirements

No.	Text
3.2.1.9	Interior steady-state noise at each crew position (driver and passengers) in the cab shall be less than XX dB (A) when PTO driven equipment, not normally utilized during vehicle movement, is not in use.
3.2.1.12.1	All complete vehicle configurations including basic chassis and cab, body assemblies, kits, cranes and ancillary equipment shall continue to function when exposed to emissions from Electromagnetic Compatibility (EMC) and Near Strike Lightning (NSL).
3.2.1.12.3	No other vehicle lighting shall be capable of being activated while in the blackout mode exceptwhere otherwise required by this ATPD.
3.2.1.14	In order to protect human health, whole body vibration shall meet the requirements of MIL-STD-1472, during testing. The vehicle shall attain no more than 6 watts average vertical absorbed power at the driver's station while negotiating a 0.7 inch Root Mean Square (RMS) course at speeds up to XX mph, a XX inch RMS course at speeds up to XX mph, and a XX inch, RMS course at speeds up to XX mph with the tires at normal cross-country inflation pressure. The vehicle shall show no more than XXg acceleration at the driver's station while negotiating half-round obstacles of XX inch height at a speed of at least XX mph, and a XX inch height at a speed of at least XX mph, with tires at normal cross-country inflation pressure.
3.2.2.1	Dimensions shall be defined in accordance with SAE J1100 except for para W103 vehicle width, which is redefined as: the maximum dimension measured between the widest points on the vehicle, excluding exterior mirrors and marker lamps, but including bumpers, moldings, and sheet metal protrusions.
3.2.2.3.3	The vehicle shall meet the requirements of the baseline XX-year corrosion prevention design of the baseline level III technical data package.
3.2.2.3.5	Dissimilar metals shall be electrically isolated to prevent galvanic corrosion
3.2.4	Each model shall have a maintenance ratio (MR) no greater than specified in Table I. REFER TO PG 16
3.2.5	Each FMTV model shall have a 0.6 probability with a 50% confidence of completing 20,000 mi. (32180 km) per the mission profile without a durability failure.
3.2.8.1	The vehicle shall meet the requirements of MIL-STD-209H, type II for helicopter transport.
3.2.8.3	The maximum height of the vehicle shall not exceed 90 in. (228 cm) for AD models.
3.2.9	The FMTV cab shall have seating provisions for three (3) crew members when radios/radio mounts are not installed, 2 crew members when installed. Doors shall comply with FMVSS 206.
3.2.9.1	When assembled, cab and all components shall be waterproof to preclude the entrance of water due to rain, melting snow, road splash and the penetration of moisture from all other causes.
3.2.9.2	The cab structure assembly shall pass a 200 hour Government approved hydropulse test to include the following installations at a minimum: entire cab structure, door locks and fittings, steering column and wheel, instrument panel array including heater and circuit breakers, wipers, washer, mirrors, all 3 seats with appropriate

	weights, machine gun ring and simulated gun mass, floor covering, drain plugs, headlights, harnesses as needed to connect everything electrical, accelerator pedal, pneumatic controls, chemical alarm and standard communications equipment, and fixed glass and seals. It shall also mount on a simulated frame including the FMTV front and rear cab mounts.
3.3.1	Radioactive materials shall not be used. All materials shall be new and unused.

After applying the pre-processing and syntax rules the following structured requirements in Table 51 were written.

Table 51: Structured FMTV cab subsystem requirements

No.	Text
3.2.1.9b	The cab shall not emit a steady-state noise level over 85 dB (A) at each crew position (driver and passengers) when PTO driven equipment, not normally utilized during vehicle movement, is not in use.
3.2.1.12.1a	The cab shall function when exposed to emissions from Electromagnetic Compatibility (EMC) and Near Strike Lightning (NSL).
3.2.1.12.1b	The body assemblies shall function when exposed to emissions from Electromagnetic Compatibility (EMC) and Near Strike Lightning (NSL).
3.2.1.12.1c	FMTV shall not produce emissions that cause Electromagnetic Interference (EMI) with mission critical equipment located within the FMTV or in the surrounding area.
3.2.1.12.3b	Vehicle lighting shall not be activated while in the blackout mode exceptwhere otherwise required by this ATPD.
3.2.1.14a	FMTV body vibration shall meet the requirements of MIL-STD-1472.
3.2.1.14b	The vehicle shall attain less than 6 watts average vertical absorbed power at the driver's station while negotiating a 0.7 inch Root Mean Square (RMS) course at speeds up to 25 mph, a 1.0 inch RMS course at speeds up to 17 mph, and a 1.5 inch, RMS course at speeds up to 12 mph with the tires at normal cross-country inflation pressure.
3.2.1.14c	The vehicle shall show no more than 2.5g acceleration at the driver's station while negotiating half-round obstacles of 8 inch height at a speed of at least 12 mph, and a 10 inch height at a speed of at least 7 mph, with tires at normal cross-country inflation pressure.
3.2.2.1	The vehicle shall have defined dimensions in accordance with SAE J1100 except for para W103 vehicle width, which is redefined as: the maximum dimension measured between the widest points on the vehicle, excluding exterior mirrors and marker lamps, but including bumpers, moldings, and sheet metal protrusions.
3.2.2.3.3	The vehicle shall meet the requirements of the baseline 22-year corrosion prevention design of the baseline level III technical data package.
3.2.2.3.5	Dissimilar metals shall be electrically isolated to prevent galvanic corrosion.
3.2.4	Each FMTV model shall have a maintenance ratio (MR) no greater than specified in Table I. REFER TO PG 16

3.2.5	Each FMTV model shall have a 0.6 probability with a 50% confidence of completing 20,000 mi. (32180 km) per the mission profile without a durability failure.
3.2.8.1c	The vehicle shall meet the requirements of MIL-STD-209H, type II for helicopter transport.
3.2.8.3	The vehicle shall have a maximum height less than 90 in. (228 cm) for AD models.
3.2.9	The FMTV cab shall have seating provisions for three (3) crew members when radios/radio mounts are not installed, 2 crew members when installed.
3.2.9.1a	Cab shall be waterproof to preclude the entrance of water due to rain, melting snow, road splash and the penetration of moisture from all other causes.
3.2.9.1b	All components shall be waterproof to preclude the entrance of water due to rain, melting snow, road splash and the penetration of moisture from all other causes.
3.2.9.2a	The entire cab structure shall pass a 200 hour Government approved hydropulse test.
3.2.9.2b	The steering wheel shall pass a 200 hour Government approved hydropulse test.
3.2.9.2c	The instrument panel array shall pass a 200 hour Government approved hydropulse test.
3.2.9.2d	The floor covering shall pass a 200 hour Government approved hydropulse test.
3.2.9.2e	The cab structure assembly shall mount on a simulated frame including the FMTV front and rear cab mounts.
3.3.1a	Radioactive materials shall not be used.
3.3.1b	All component materials shall be new and unused.

These cab subsystem requirements refer not only to the leaf node parts of the physical design (components) but to the branch as well (cab subsystem).

Step 2: Map Requirements to Components

The component and assemblies of the cab subsystem are obtained from CAD models. The component information is summarized in Table 52.

Table 52: FMTV cab subsystem component list

Component	Figure	Mass (kg)
Steering Wheel	Figure 86	1.00
Instrument Panel	Figure 87 Figure 88	68.61
Cab Housing	Figure 89	356.84
Cab Floors	Figure 90	289.29

These cab subsystem requirements refer not only to leaf node parts of the physical design (components), but to the branch as well (cab subsystem). Hence, a hierarchy is represented in the requirement list. By Rule 1 of the requirement preprocessing rules, the subject of the requirement has to be part of the physical subsystem. This physical subsystem is a hierarchy also and is modeled in MagicDraw SysML and is included in Figure 59.

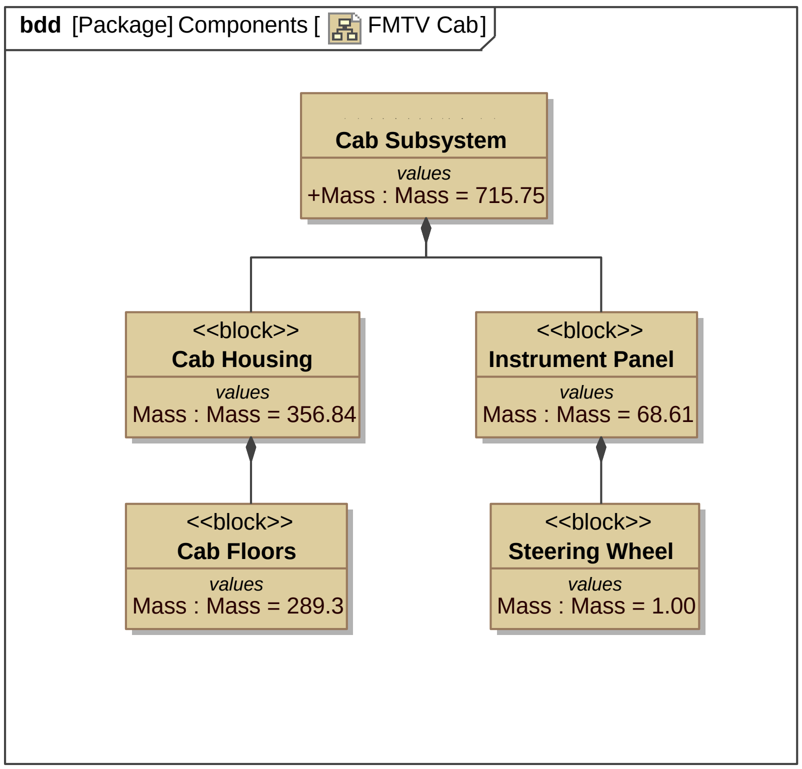


Figure 59: FMTV cab subsystem

The component hierarchy is used in populating the RxC matrix when the subject of the requirement is something other than a leaf node of the hierarchy, like the FMTV cab subsystem. A requirement with the subject as the FMTV cab subsystem will be related to all components that are in that subsystem, all the leaf nodes that comes from that branch of the hierarchy. Once the components have been acquired, the requirements are related to the component hierarchy. An example of a cab requirement related to the component hierarchy is shown in Figure 60.

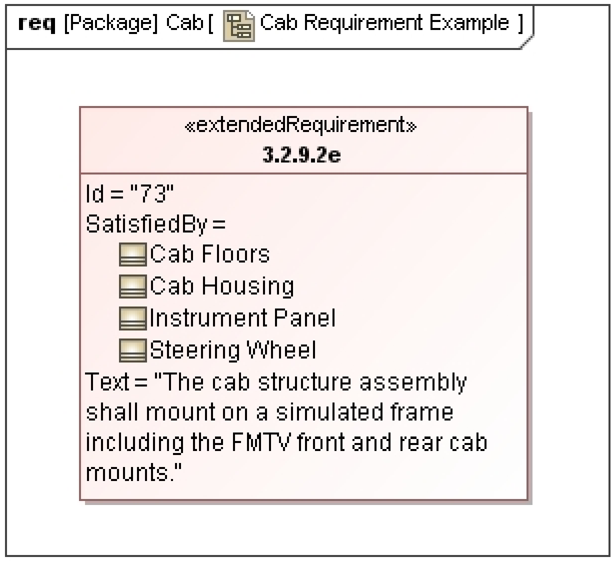


Figure 60: Example SysML FMTV cab subsystem requirement with component relationships

The requirement 3.2.9.2e (refer to Table 41 for the requirement text) is related to the Frame. A DSM matrix of requirements and components is then constructed in the SysML software and is shown in Figure 61.

	Cab Floors [Components ::Cab]	Cab Housing [Components ::Cab]	Instrument Panel [Components ::Cab]	Steering Wheel [Components ::Cab]
Cab [Requirements]	18	17	17	16
54 3.2.1.12.1a [Requireme…	↗	↗	↗	↗
55 3.2.1.12.1b [Requireme…	↗	↗	↗	↗
56 3.2.1.12.1c [Requireme…	↗	↗	↗	↗
57 3.2.1.12.3b [Requireme…			↗	
58 3.2.1.14a [Requirement…	↗	↗		
59 3.2.1.14b [Requirement…	↗	↗	↗	↗
60 3.2.1.14c [Requirement…	↗	↗	↗	↗
53 3.2.1.9b [Requirements…	↗	↗	↗	↗
61 3.2.2.1 [Requirements::…	↗	↗	↗	↗
62 3.2.2.3.3 [Requirement…	↗	↗	↗	↗
63 3.2.4 [Requirements::Cab]	↗	↗	↗	↗
64 3.2.5 [Requirements::Cab]	↗	↗	↗	↗
65 3.2.8.1c [Requirements:…	↗	↗	↗	↗
66 3.2.9 [Requirements::Cab]	↗	↗	↗	↗
67 3.2.9.1a [Requirements…	↗	↗	↗	↗
68 3.2.9.1b [Requirements…	↗	↗	↗	↗
69 3.2.9.2a [Requirements…	↗	↗		
70 3.2.9.2b [Requirements…				↗
71 3.2.9.2c [Requirements:…			↗	
72 3.2.9.2d [Requirements…	↗			
73 3.2.9.2e [Requirements…	↗	↗	↗	↗
Shared [Requirements]	4	4	4	4
46 3.2.2.3.5 [Requirement…	↗	↗	↗	↗
47 3.2.8.3 [Requirements::…	↗	↗	↗	↗
48 3.3.1a [Requirements::…	↗	↗	↗	↗
6 3.3.1b [Requirements::S…	↗	↗	↗	↗

Figure 61: SysML DSM matrix of FMTV cab subsystem

The SysML DSM is exported to Excel to be used to construct the RxR matrices. The exported matrix in Excel represents relationships with non-numbers. These are changed to either binary or mass strengths. The two DSMs are shown in Table 53 and Table 54.

Table 53 : FMTV cab subsystem RxC binary matrix

	steering wheel	instrument panel	cab housing	cab floors
3.2.1.9b	1	1	1	1
3.2.1.12.1a	1	1	1	1
3.2.1.12.1b	1	1	1	1
3.2.1.12.1c	1	1	1	1
3.2.1.12.3b	0	1	0	0
3.2.1.14a	0	0	1	1
3.2.1.14b	1	1	1	1
3.2.1.14c	1	1	1	1
3.2.2.1	1	1	1	1
3.2.2.3.3	1	1	1	1
3.2.2.3.5	1	1	1	1
3.2.4	1	1	1	1
3.2.5	1	1	1	1
3.2.8.1c	1	1	1	1
3.2.8.3	1	1	1	1
3.2.9	1	1	1	1
3.2.9.1a	1	1	1	1
3.2.9.1b	1	1	1	1
3.2.9.2a	0	0	1	1
3.2.9.2b	1	0	0	0
3.2.9.2c	0	1	0	0
3.2.9.2d	0	0	0	1
3.2.9.2e	1	1	1	1
3.3.1a	1	1	1	1
3.3.1b	1	1	1	1

Table 54: FMTV cab subsystem RxC mass matrix

	steering wheel	instrument panel	cab housing	cab floors
	1.00	68.61	356.835	289.291
3.2.1.9b	1.00	68.612	356.835	289.291
3.2.1.12.1a	1.00	68.612	356.835	289.291
3.2.1.12.1b	1.00	68.612	356.835	289.291
3.2.1.12.1c	1.00	68.612	356.835	289.291
3.2.1.12.3b	0	68.612	0	0
3.2.1.14a	0	0	356.835	289.291
3.2.1.14b	1.00	68.612	356.835	289.291
3.2.1.14c	1.00	68.612	356.835	289.291
3.2.2.1	1.00	68.612	356.835	289.291
3.2.2.3.3	1.00	68.612	356.835	289.291
3.2.2.3.5	1.00	68.612	356.835	289.291
3.2.4	1.00	68.612	356.835	289.291
3.2.5	1.00	68.612	356.835	289.291
3.2.8.1c	1.00	68.612	356.835	289.291
3.2.8.3	1.00	68.612	356.835	289.291
3.2.9	1.00	68.612	356.835	289.291
3.2.9.1a	1.00	68.612	356.835	289.291
3.2.9.1b	1.00	68.612	356.835	289.291
3.2.9.2a	0	0	356.835	289.291
3.2.9.2b	1.00	0	0	0
3.2.9.2c	0	68.612	0	0
3.2.9.2d	0	0	0	289.291
3.2.9.2e	1.00	68.612	356.835	289.291
3.3.1a	1.00	68.612	356.835	289.291
3.3.1b	1.00	68.612	356.835	289.291

The RxC binary matrix shows the existence of relationships while the RxC mass matrix shows a weighted relationship. To create the RxR mass matrix, the RxC mass matrix is multiplied with the transpose of the RxC binary matrix. For illustration purposes, the transpose is displayed in Table 55.

Table 55: FMTV cab subsystem CxR matrix

	3.2.1.9b	3.2.1.12.1a	3.2.1.12.1b	3.2.1.12.1c	3.2.1.12.3b	3.2.1.14a	3.2.1.14b	3.2.1.14c	3.2.2.1	3.2.2.3.3	3.2.2.3.5	3.2.4	3.2.5	3.2.8.1c	3.2.8.3	3.2.9	3.2.9.1a	3.2.9.1b	3.2.9.2a	3.2.9.2b	3.2.9.2c	3.2.9.2d	3.2.9.2e	3.3.1a	3.3.1b
steering wheel	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1
instrument panel	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1
cab housing	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
cab floors	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1

The equation used to create the RxR mass matrix for the FMTV cab subsystem is shown below.

$$R \times C\,Mass \;\times\; C \times R\,Mass \;=\; R \times R\,Mass \qquad \text{Equation 8.1}$$

The RxR mass matrix shows

3. The amount of mass each requirement affects (diagonal)
4. The amount of mass affected by two requirements (off-diagonal)

The resulting RxR mass matrix is shown in Table 56. Due to the very large size of the table, only an excerpt is shown.

Table 56: Excerpt of FMTV cab subsystem RxR mass matrix

	3.2.1.9	3.2.1.12.1a	3.2.1.12.1b	3.2.1.12.1c	3.2.1.12.3b	3.2.1.14a	3.2.1.14b
3.2.1.9b	716	716	716	716	69	646	716
3.2.1.12.1a	716	716	716	716	69	646	716
3.2.1.12.1b	716	716	716	716	69	646	716
3.2.1.12.1c	716	716	716	716	69	646	716
3.2.1.12.3b	69	69	69	69	69	0	69
3.2.1.14a	646	646	646	646	0	646	646
3.2.1.14b	716	716	716	716	69	646	716
3.2.1.14c	716	716	716	716	69	646	716
3.2.2.1	716	716	716	716	69	646	716
3.2.2.3.3	716	716	716	716	69	646	716
3.2.2.3.5	716	716	716	716	69	646	716
3.2.4	716	716	716	716	69	646	716
3.2.5	716	716	716	716	69	646	716
3.2.8.1c	716	716	716	716	69	646	716
3.2.8.3	716	716	716	716	69	646	716
3.2.9	716	716	716	716	69	646	716
3.2.9.1a	716	716	716	716	69	646	716

The RxR binary matrix was created by multiplying the RxC binary matrix with the CxR binary matrix. This matrix shows

3. The number of components each requirement affects (diagonal)
4. The number of components affected by two requirements (off-diagonal)

The FMTV cooling subsystem RxR binary matrix is shown in Table 57.

Table 57: FMTV cab subsytem RxR binary matrix

RxR_1/0	3.2.1.9b	3.2.1.12.1a	3.2.1.12.1b	3.2.1.12.1c	3.2.1.12.3b	3.2.1.14a	3.2.1.14b	3.2.1.14c	3.2.2.1	3.2.2.3.3	3.2.2.3.5	3.2.4	3.2.5	3.2.8.1c	3.2.8.3	3.2.9	3.2.9.1a	3.2.9.1b	3.2.9.2a	3.2.9.2b	3.2.9.2c	3.2.9.2d	3.2.9.2e	3.3.1a	3.3.1b
3.2.1.9	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.1.12.1a	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.1.12.1b	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.1.12.1c	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.1.12.3b	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1
3.2.1.14a	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	1	2	2	2
3.2.1.14b	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.1.14c	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.2.1	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.2.3.3	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.2.3.5	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.4	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.5	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.8.1c	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.8.3	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.9	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.9.1a	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.9.1b	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.2.9.2a	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	1	2	2	2
3.2.9.2b	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1
3.2.9.2c	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1
3.2.9.2d	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
3.2.9.2e	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.3.1a	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4
3.3.1b	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4	4	4	2	1	1	1	4	4	4

Step 3: Requirement Analysis and Identification of Mass Intensive Requirements

Using the RxR binary and mass matrices, the total mass affected and the coupling levels were attained and compiled in Table 58.

Table 58: Requirement mass and coupling data for FMTV cab subsystem

	total mass affected by 1 requirement	total mass coupled to (w/other requirements)	# requirements related to
3.2.1.9b	716	14603	25
3.2.1.12.1a	716	14603	25
3.2.1.12.1b	716	14603	25
3.2.1.12.1c	716	14603	25
3.2.1.12.3b	69	1372	21
3.2.1.14a	646	13212	22
3.2.1.14b	716	14603	25
3.2.1.14c	716	14603	25
3.2.2.1	716	14603	25
3.2.2.3.3	716	14603	25
3.2.2.3.5	716	14603	25
3.2.4	716	14603	25
3.2.5	716	14603	25
3.2.8.1c	716	14603	25
3.2.8.3	716	14603	25
3.2.9	716	14603	25
3.2.9.1a	716	14603	25

3.2.9.1b	716	14603	25
3.2.9.2a	646	13212	22
3.2.9.2b	1	19	20
3.2.9.2c	69	1372	21
3.2.9.2d	289	6075	22
3.2.9.2e	716	14603	25
3.3.1a	716	14603	25
3.3.1b	716	14603	25

The values in the column showing "total mass affected by 1 requirement" are calculated by taking the diagonal values for each requirement from the RxR mass matrix in Table 46. The values in the column showing "total mass coupled to (w/ other requirements)" are calculated by taking the sum of the rows in the RxR mass matrix in Table 46 and subtracting the value on the diagonal. This shows the amount of mass the selected requirement and all other requirements coupled to it affect. The values in the column showing "# requirements related to" shows the coupling of each requirement. They are found by counting the number of nonzero values in each row of the RxR binary matrix.

Taking into account mass and coupling leads to several different combination possibilities: requirements that affect much mass and are also highly coupled, requirements that affect much mass and are lowly coupled, requirements that affect little mass and are highly coupled and requirements that affect little mass and are lowly coupled. Also to be included are mid-level mass or coupling values. These are always treated as second-choice options if the best case option is not available. This is graphically shown in Figure 62.

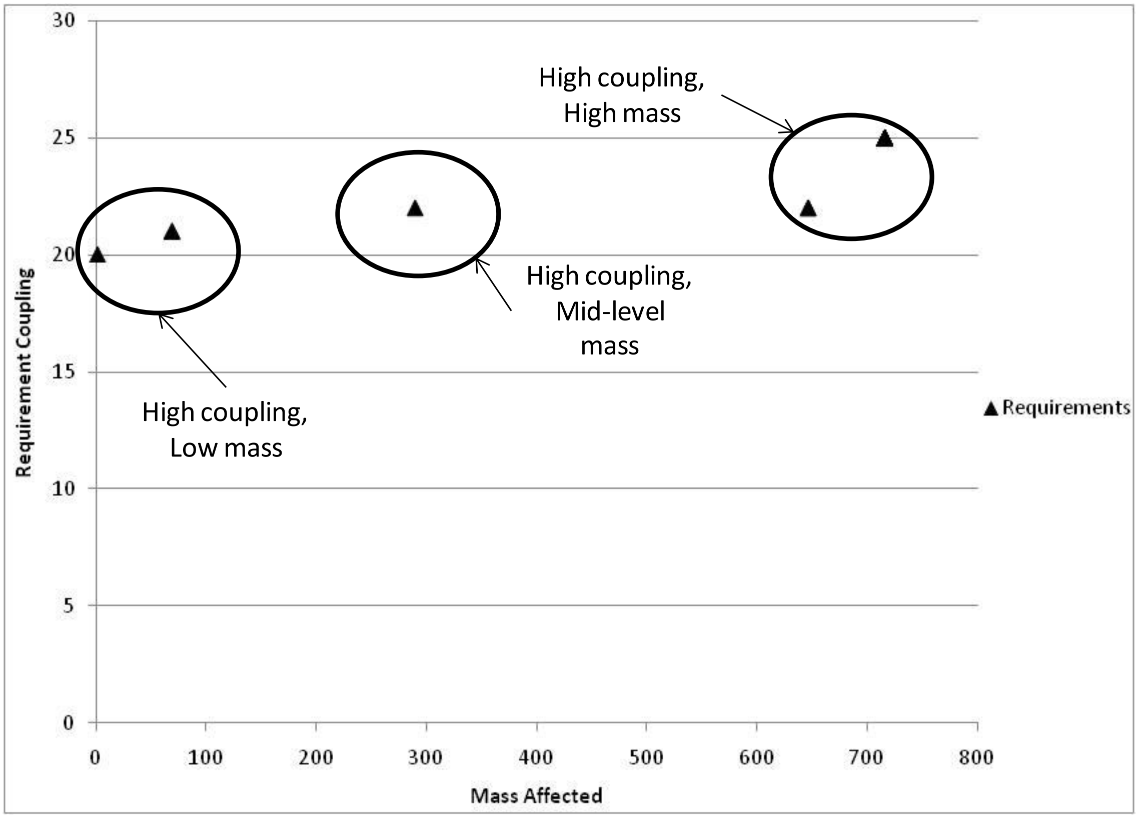


Figure 62: Requirement coupling vs. mass for FMTV cab subsystem

This figure shows the cab requirements fit into three categories: high coupling high mass, high coupling mid-level mass and high coupling low mass. Each data point stands for the group of requirements that have the same values for requirement mass and coupling. For example, the second data point from the high coupling low mass data point represents requirements 3.2.1.12.3b and 3.2.9.2c from the requirements list which both have mass values of 69 and coupling values of 21.

Given this discussion of how requirements are selected to change in Chapter 3, the following Table 59 shows the order in which requirements should be modified to most efficiently reduce mass.

Table 59: Order priority for which requirements to change of FMTV cab subsystem

	total mass affected by 1 requirement	total mass coupled to (w/other requirements)	# requirements related to
3.2.1.9b	716	14603	25
3.2.1.12.1a	716	14603	25
3.2.1.12.1b	716	14603	25
3.2.1.12.1c	716	14603	25
3.2.1.14b	716	14603	25
3.2.1.14c	716	14603	25
3.2.2.1	716	14603	25
3.2.2.3.3	716	14603	25
3.2.2.3.5	716	14603	25
3.2.4	716	14603	25
3.2.5	716	14603	25
3.2.8.1c	716	14603	25
3.2.8.3	716	14603	25
3.2.9	716	14603	25
3.2.9.1a	716	14603	25
3.2.9.1b	716	14603	25
3.2.9.2e	716	14603	25
3.3.1a	716	14603	25
3.3.1b	716	14603	25
3.2.1.14a	646	13212	22
3.2.9.2a	646	13212	22
3.2.9.2d	289	6075	22
3.2.9.2c	69	1372	21
3.2.1.12.3b	69	1372	21
3.2.9.2b	1	19	20

Requirements that affect high mass and are highly coupled are changed first while requirements that affect little mass and are lowly coupled are changed last.

Chapter 9. Closure

Discussion

Prioritizing requirements in subsystems

The proposed requirement analysis method is used to compile an ordered list of requirements to reduce mass for three FMTV subsystems. This list is shown in Table 39 for the cooling subsystem, Table 49 for the chassis subsystem and Table 59 for the cab subsystem. Ranking of requirements to change does not need to be constrained only to single subsystems, however. Subsystems can be ranked as being better able to reduce mass than other subsystems. Consider Figure 63.

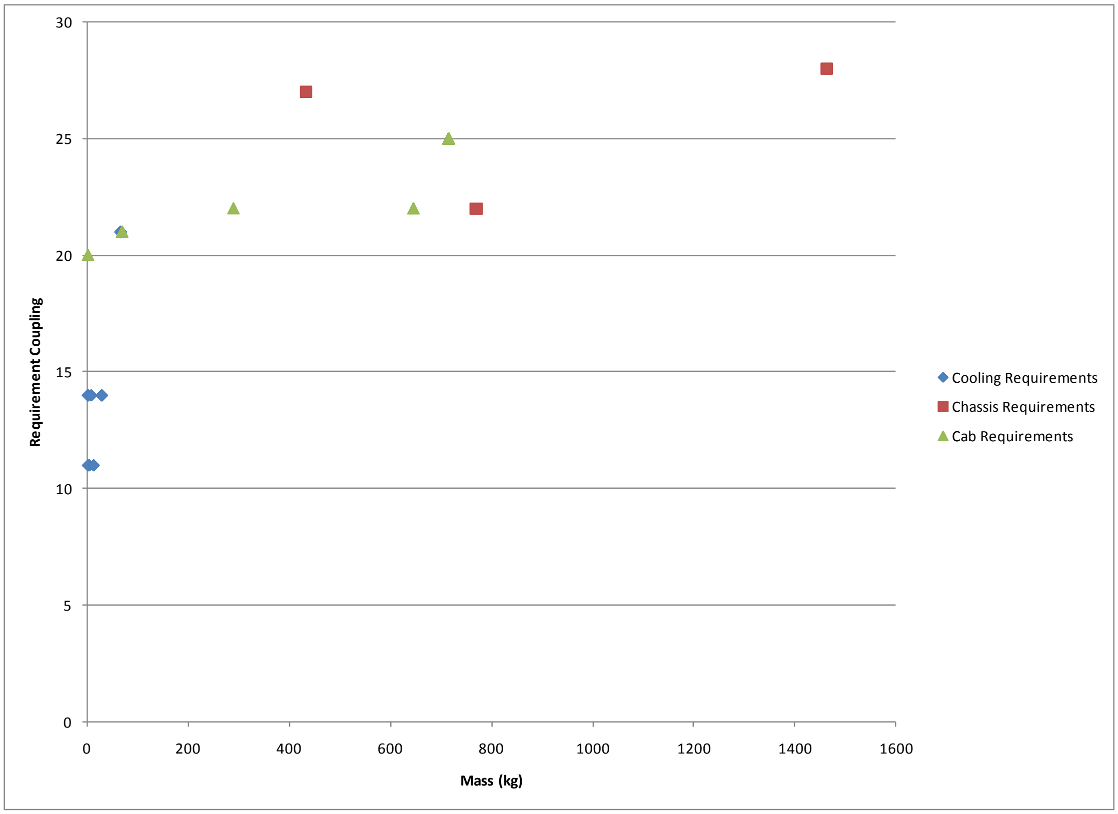


Figure 63: Requirements of three FMTV subsystems plotted comparing mass to coupling

This figure shows that requirements for the cooling subsystem are least coupled compared to the other subsystems, but they also affect the least amount of mass. The cab subsystem affects greater mass that the cooling subsystem, but cab requirements also have higher coupling values. The chassis requirements have the highest mass values but they also have the highest coupling values. Decreasing the requirement coupling decreases the amount of mass affected. Increasing the mass affected increases the amount of requirement coupling. A tradeoff has to be made between mass and coupling. The cab and chassis subsystems are highly coupled, with requirements coupled to at least 20 other requirements. The only viable alternative is to change requirements in the cooling subsystem first even though it affects significantly less mass than the other two subsystems. The mass values for select requirements in the cooling subsystem (displayed in Table 38) are as high as 65 kg. This allows us to not only prioritize which requirements to change in a subsystem, but to prioritize which subsystems to change first.

Analysis of Processed Rules

The processed requirement list was analyzed to show the number of times each pre-processing rule was used. Some requirements needed only one rule applied to them while others needed several applied. The statistics are presented in Table 60.

Table 60: Statistics for number of times and combinations of pre-processing rules were used

Rules Used	# Times	
1,2,3	1	0.22%
1,2,5,8	1	0.22%
1,2,7	1	0.22%
1,2	5	1.08%
1,3,4	1	0.22%
1,3	2	0.43%
1,4	1	0.22%
1,5,10	1	0.22%
1,5	7	1.51%
1,7	1	0.22%
1,9	1	0.22%
1,10	3	0.65%
1	51	11.02%
2,3,10	1	0.22%
2,3	8	1.73%
2,10	2	0.43%
2	35	7.56%
3,5	2	0.43%
3,10	4	0.86%
3	47	10.15%
4	4	0.86%
5	40	8.64%
6	2	0.43%
7,8	1	0.22%
7	1	0.22%
8	1	0.22%
9,10	1	0.22%
9	1	0.22%
10	16	3.46%
None	222	47.95%

Notice that almost half of the requirements were already stated according to the pre-processing rules. Of the other half of the requirements, while combinations of rules were used, the most of requirements that were changed used single rules. There were 198 requirements that used single rules of the total 464 requirements (222 of the requirements did not use any rules). Another way to view the data would be to consider the number of times each rule was used in a requirement, either by itself or in combination with other rules. Table 61 shows this data.

Table 61: Percentage of the time each requirement was used in the total requirement list

Rule 1	16.41%
Rule 2	11.66%
Rule 3	14.25%
Rule 4	1.30%
Rule 5	11.02%
Rule 6	0.43%
Rule 7	0.86%
Rule 8	0.65%
Rule 9	0.65%
Rule 10	6.05%
None	47.95%

Note that Rules 1,2,3 and 5 were used for half of the changed requirements. Rules 1,2 and 3 are also the most important rules to use since they allow the requirements to be related in relational matrices.

It is recommended to the Army that rules 1,2 and 3 be used at least in future requirement documentation. These rules have been shown to be the most important to the proposed analysis method and also are used the most in the requirement standardization. The other rules deal with the placement of description phrases within the requirement. These phrases are not addressed or used in the proposed requirement method and can therefore be placed anywhere in the statement. As mentioned in Chapter 3, these phrases should be placed consistently within the requirement statement.

Key Contributions and Limitations

Early on in this thesis, it was discovered that if consistent matrices were to be constructed between designers, that is, in order for designers to consistently create agreeing relational matrices based on natural language requirements, the natural language requirement statement must be stated consistently. This conclusion led to the formulation of ten requirement pre-processing rules in Chapter 3. These rules dictate the grammar of the requirement sentence addressing the subject, verb and adjunct phrase. These rules were applied to three FMTV subsystems and the correctly stated requirements are shown in Table 31 for the cooling subsystem, Table 41 for the chassis subsystem and Table 51 for the cab subsystem. The FMTV requirements as given from the Army numbered 128. These requirements were decomposed into 754 consistently stated requirements.

Relational matrices, particularly DSMs and DMMs were used significantly in this thesis to generate the data used to prioritize requirements. Manipulating DSMs and DMMs are accomplished mathematically, eliminating the concern of varying performance between different designers. Identifying requirements to change is accomplished by setting rules (by sorting) the mass and coupling data for each requirement.

By standardizing the way requirements are stated using pre-processing rules and syntax, a significant portion of the proposed requirement analysis method is automatable. The only exception being when relating the requirement subject to leaf nodes (components) when the subject of the sentence is a branch (subsystem) to the component list in a DSM RxC matrix. The designer must establish the component hierarchy.

A limitation of this thesis is the bottom-up direction of the proposed requirement analysis method. This method reverse engineers existing FMTV subsystems. A correction of this limitation is addressed in the second research question in the future works section.

Another limitation of this method is that only a maximum mass value can be mapped to each requirement. This method cannot map an exact value to each requirement showing how much mass it will affect if changed. Changing requirements in different ways may also change the amount of mass one requirement affects. Thus, one requirement may affect a varying amount of mass by varying how the requirement changes.

Validation

The research question addressed in this thesis is shown below.

> How can requirements can be related to mass in the early part of the design process, in the design specification (requirements) phase?

A proposed requirement analysis method was developed that answers this question in the affirmative. Three example problems were given to demonstrate the method. In this section a deeper validation of the method is presented. A validation square is used to show the validation of this method in this thesis.

Table 62: Illustration of the validation square [6]

(1) and (2) Theoretical and Structural Validity	(6) Theoretical Performance Validity
(3) Empirical Structural Validity	(4) and (5) Empirical Performance Validity

Part (1) was accomplished using an extensive literature review in Chapter 2. The four primary constructs used in this method are requirement capabilities, relational matrices, requirement rules and requirement syntax rules. The two constructs requirement capabilities and relational matrices are well known and have extensive literature discussing them. Requirement capabilities are discussed in Chapter 2 while relational matrices are discussed in Chapter 3. Requirement capabilities are included in Table 2 with sources showing each requirement capability used in other literature. Method consistency was addressed in **Part (2)** by using flowcharts to show the information flow within a method [6]. This is accomplished in Chapter 3 when the proposed requirement analysis method is introduced by using a flowchart the illustration in Figure 7. The example problems are shown in Chapter 6, Chapter 7 and Chapter 8 with three subsystem examples to show the empirical structural validity in **part (3)**. The outcome of the method was shown in **part (4).** The results attained do indeed answer the research challenge of being able to map requirements to mass. The results of relating requirements to mass are shown for the cooling subsystem in Table 39, Table 49 for the chassis subsystem and Table 59 for the cab subsystem. **Part (5)** was shown by explaining how each part of the method significantly contributes to the results attained from the method. This was shown throughout Chapter 3 in the introduction and discussion of the method. Preprocessing rules were needed to relate requirements to components, relational matrices were needed to relate requirements to components and to each other and the SysML software MagicDraw is used to maintain the requirement and component entities, the relationships between them and the relational matrices. **Part (6)** involves showing the usefulness of the method beyond the example problems [6].

Future Work

Future work for this research can come in two areas. The first area is in step 1 of the requirements analysis process, completely automating the preprocessing rules/requirements. The future research question for step 1 would be:

> What techniques can be used to enable automated analysis and/or real-time guidance of engineering requirements during the elicitation process?

In this research, requirement correction is accomplished by hand and can be a painstaking process depending on the length of the requirement document. The parts of speech are manually identified, the requirement rules that apply are manually identified and the corrections to the requirements are manually made. To automate the requirement correction would be a substantial improvement over the manual one. To accomplish this, a part of speech (POS) tagger could be used to identify the parts of speech in the requirement. This is done using a vocabulary for each part of speech to identify which words can be used as subjects, verbs, objects and modifiers. For example, the word ‘vehicle’ would be identified as the subject since the word is included in the subject vocabulary. One challenge of this future work is the length of the vocabulary for each of the parts of speech. The same word could not be included in multiple vocabularies except if there was a way to identify the part of the requirement it is located in. For instance, ‘vehicle’ could also be used as the direct object of the requirement. This future work would depend completely, however, on structuring the requirement statement in a specific way.

The second area of future research addresses steps 2,3 and 4 of the requirement analysis process.

> How can engineering requirements and mass analysis be supported through a top-down approach while the system architecture and components are not established?

This requirement analysis method for relating mass to requirements is a bottom up method requiring existing components to be known; it is not top down. It only uses requirements with knowledge of the existing system. In order to create a top down method, it is proposed that this requirement analysis method be applied to many designs and the mass intensive requirements be examined. Requirements can be classified into different types. According to Paul and Beitz [7], requirements can be classified into categories such as safety, energy, assembly, costs, recycling and geometry, just to name a few. If the mass intensive requirements are examined with requirement types in mind, it might be found that certain categories of requirements are more mass intensive than others. This knowledge could be used on new designs with unknown components to identify requirements by type and to identify the requirements that could possibly be mass intensive. Another facet of this future work would be to apply this method to different types of designs such as aerospace, naval or automotive and see if the same types of requirements are identified across these fields.

Bibliography

[1] TACOM, (1998), "Tactical Vehicle Fleetbook", *Technical Report*, U.S Army Tank-automotive and Armaments Command (TACOM), Washington D.C.

[2] Board, D. S., (2001), "Report of the Defense Science Board on More Capable Warfighting through Reduced Fuel Burden", *Technical Report*, Office of the Under Secretary of Defense for Acquisition and Technology, Washington D.C.

[3] Nygren, K. P., Massie, D. D., and Kern, P. J., "Army Energy Strategy for the End of Cheap Oil", *Technical Report*, United States Military Academy, West Point, NY.

[4] Hirsch, R. L., Bezdek, R. H., and Wendling, R. M., (2005), "Peaking of World Oil Production: Sooner Rather Than Later", *Issues in Science and Technology*, **XXI**(**3**), 25-30.

[5] Piggee, A. F., (2002), "Transformation-Revolution in Military Logistics", *Technical Report*, US Army War College, Carlisle Barracks, Pa.

[6] Pedersen, K., Emblemsvag, J., Bailey, R., Allen, J. K., and Mistree, F., (2000), "Validating Design Methods and Research: The Validation Square", *ASME Design Engineering Technical Conferences*, Baltimore, Maryland,

[7] Pahl, G., and Beitz, W., (2007), *Engineering Design*, Springer-Verlag, London.

[8] Ullman, D. G., (2003), *The Mechanical Design Process*, McGraw-Hill, New York.

[9] Shigley, J. E., (1977), *Series in Mechanical Engineering*, McGraw-Hill.

[10] Blanchard, B. S., and Fabrycky, W. J., (1981), *Systems Engineering and Analysis*, Prentice-Hall, Englewood.

[11] Suh, N. P., (2001), *Axiomatic Design: Advances and Applications*, Oxford University Press, New York.

[12] Dym, C. L., and Little, P., (1999), *Engineering Design*, JohnWiley & Sons, New York.

[13] Sud, R. R., and Arthur, J. D., *Requirements Management Tools: A Qualitative Assessment*.

[14] Mylopoulos, J., Chung, L., Liao, S., Wang, H., and Yu, E., (2001), "Exploring Alternatives During Requirements Analysis", *IEEE Software*, **2001**(**January/February**).

[15] Gotel, O. C. Z., and Finkelstein, A. C. W., (1994), "An Analysis of the Requirements Traceability Problem", *IEEE*, 94-101.

[16] Hoffmann, M., Kuhn, N., Weber, M., and Bittner, M., (2004), "Requirements for Requirements Management Tools", *Proceedings of the 12th IEEE International Requirements Engineering Conference*, 8,

[17] Ramesh, B., (1998), "Factors Influencing Requirements Traceability Practice", *Communications of the ACM*, **41**(**12**), 37-44.

[18] McLellan, J. M., Maier, J. R. A., Fadel, G. M., and Mocko, G., (2009), "A Method for Identifying Requirements Critical to Mass Reduction", *International Design Structure Matrix Conference*, Greenville,

[19] Zave, P., and Jackson, M., (1997), "Four Dark Corners of Requirements Engineering", *ACM Transactions on Software Engineering and Methodology*, **6**(**1**), 1-30.

[20] McLellan, J. M., Maier, J. R. A., Fadel, G. M., and Mocko, G., (2009), "Generating Dsm and Dmm Matrices Using Sysml", *Design Structure Matrix Conference*, Greenville,

[21] Dorfman, M., (1997), "Requirements Engineering", *Software Requirements Engineering*.

[22] Wiegers, K. E., (1999), "First Things First: Prioritizing Requirements", *Software Development*.
[23] Gervasi, V., and Nuseibeh, B., (2000), "Lightweight Validation of Natural Language Requirements; a Case Study", *4th International Conference on Requirements Engineering*, Pisa, 140-148,
[24] Friedenthal, S., Moore, A., and Steiner, R., (2008), *A Practical Guide to Sysml*, Morgan Kaufmann Publishers, Burlington.
[25] Karlsson, J., and Ryan, K., (1997), "A Cost-Value Approach for Prioritizing Requirements", *IEEE Software*.
[26] Hull, E., Jackson, K., and Dick, J., (2005), *Requirements Engineering*, Springer, London.
[27] Davis, A., (1993), *Software Requirements: Objects, Functionsand States*, Prentice Hall.
[28] Easterbrook, S., and Nuseibeh, B., (1995), "Using Viewpoints for Inconsistency Management", *IEEE Software Engineering Journal*, November.
[29] Chen, L., and Zeng, Y., (2009), "Automatic Generation of Uml Diagrams from Product Requirements Described by Natural Language", *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, San Diego, California,
[30] Soares, M. S., and Vrancken, J., (2007), "Requirements Specification and Modeling through Sysml", *Technical Report*, Delft University of Technology.
[31] Chomsky, N., (1965), *Aspects of the Theory of Syntax*, M.I.T. Press, Cambridge.
[32] Lamar, C., and Mocko, G. M., (2010), "Linguistic Analysis of Natural Language Engineering Requirement Statements", *Tools and Methods for Competitive Engineering*, Ancona, Italy,
[33] Chitchyan, R., Rashid, A., Rayson, P., and Waters, R., (2007), "Semantics-Based Composition for Aspect-Oriented Requirements Engineering", *Proceedings of the 6th international conference on Aspect-oriented software development*, Vancouver, 208.
[34] Jacky, J., (1997), *The Way of Z*, Cambridge University Press, Cambridge.
[35] Weilkiens, T., (2006), *Systems Engineering with Sysml/Uml: Modeling, Analysis, Design*, Morgan Kauffman Publishers, Burlington.
[36] (1977), *Requirements Engineering*, IEEE Computer Society Press, Los Alamitos, California.
[37] Clark, D. L., Beaty, J. O., Bowyer, J. W., and Neu, J. L., (1935), *Form and Style: A Manual of Composition and Rhetoric*, F. S. Crofts and Co., New York.
[38] Mylopoulos, J., Chung, L., and Nixon, B., (1992), "Representing and Using Nonfunctional Requirements: A Process-Oriented Approach", *IEEE Transactions on Software Engineering*, 18(6).
[39] Defense, D. o., (2003), "Mil-Std-961e. Dod Standard Practice", D. o. Defense, ed.
[40] Berk, L. M., (1999), "English Syntax", *English Syntax: From Word to Discourse*, Oxford University Press, New York.
[41] Browning, T. R., (2001), "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions", *IEEE Transactions on Engineering Management*, 48(3).
[42] Danilovic, M., and Browning, T. R., (2007), "Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices", *International Journal of Project Management*, 25(3), 300-314.
[43] Blaha, M., and Rumbaugh, J., (2005), *Object-Oriented Modeling and Design with Uml*, Pearson Prentice Hall, Upper Saddle River.
[44] MagicDraw, (2009), "Magicdraw User's Manual".

[45] Osenga, (2003), "S&S Wins Huge Army Truck Contract: First Rebid since S&S Won 1991 Cont4act; Potentially 23,000 2.5 Adn 5.0 Ton Trucks over Five Years", *Diesel Progress North American Edition*.

[46] (2009), "Fmtv 2010-2015: Oshkosh Wins the Re-Compete", Defense Industry Daily: Military Purchasing News for Defense Procurement Managers and Contractors.

[47] Systems, B., (2009), "Fmtv A1 Rm1080 A1 - 2.5 Ton Chassis", *Technical Report*, Sealy, Texas.

[48] Systems, B., (2009), "Fmtv A1 Rm1092 A1 - 5.0 Ton Chassis", *Technical Report*, Sealy, Texas.

Appendix 1: Processed FMTV Requirements

Due to the nature of the requirements processed in this thesis, they have been removed for publication. Please contact Professor Gregory M. Mocko at gmocko@clemson.edu for more information.

Appendix 2: FMTV Subsystem/Component Pictures

FMTV Engine Cooling Subsystem

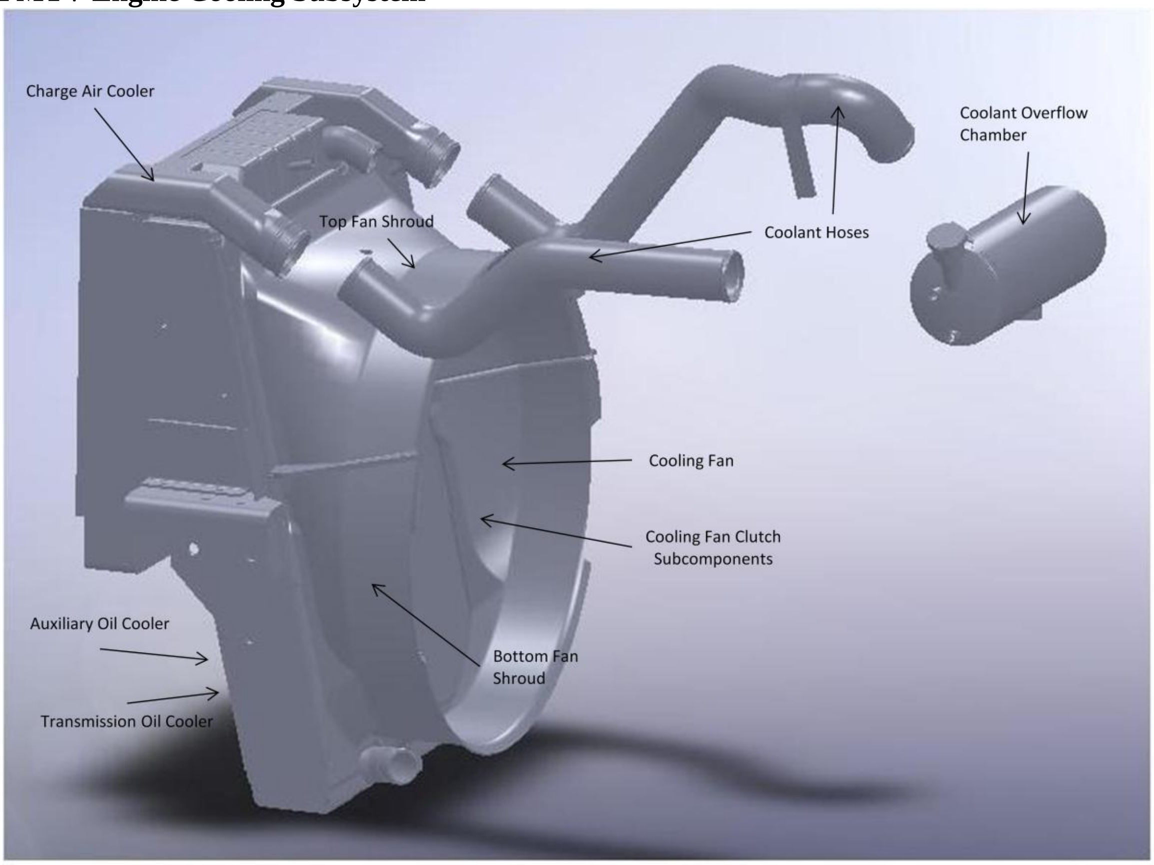


Figure 64: Cooling System Subsystem

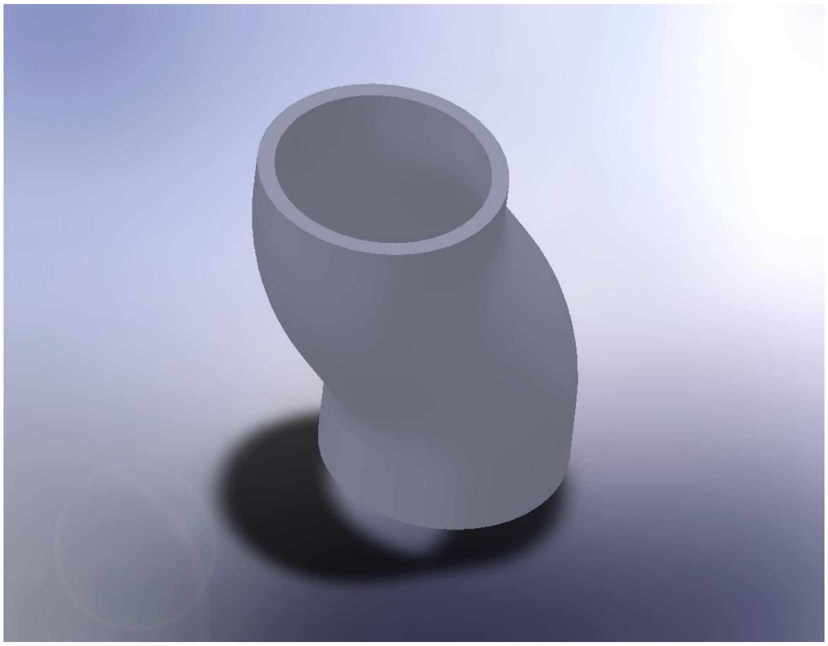


Figure 65: Cooling System Coolant Hoses

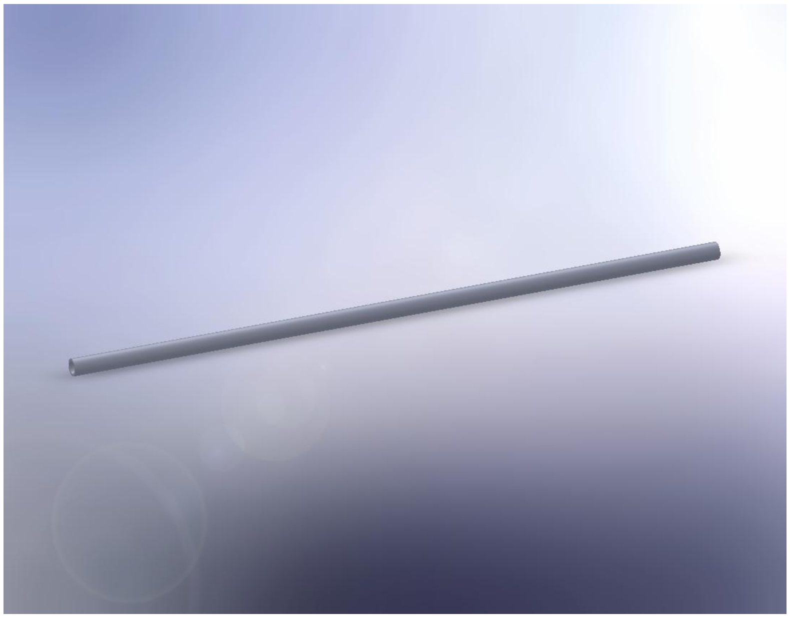


Figure 66: Cooling System Coolant Hoses

Figure 67: Cooling System Coolant Hoses

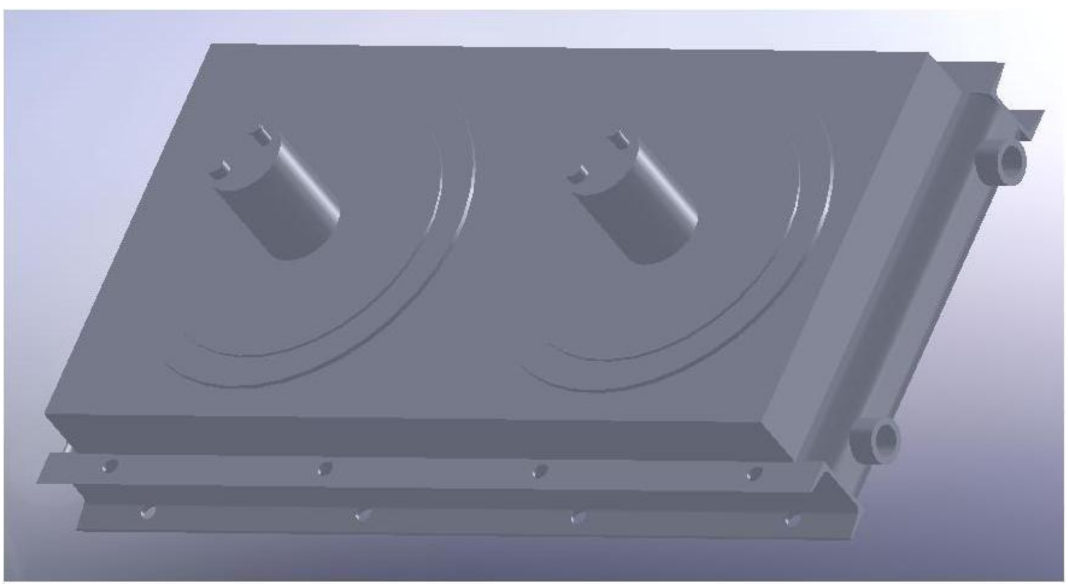


Figure 68: Cooling System Transmission Oil Cooler

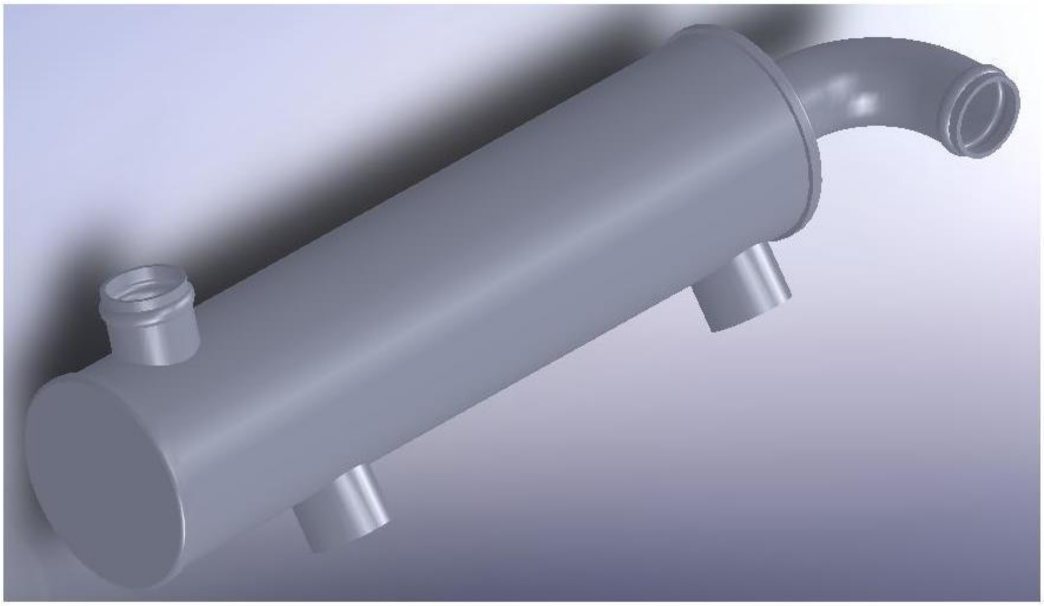


Figure 69: Cooling System Coolant Overflow Chamber

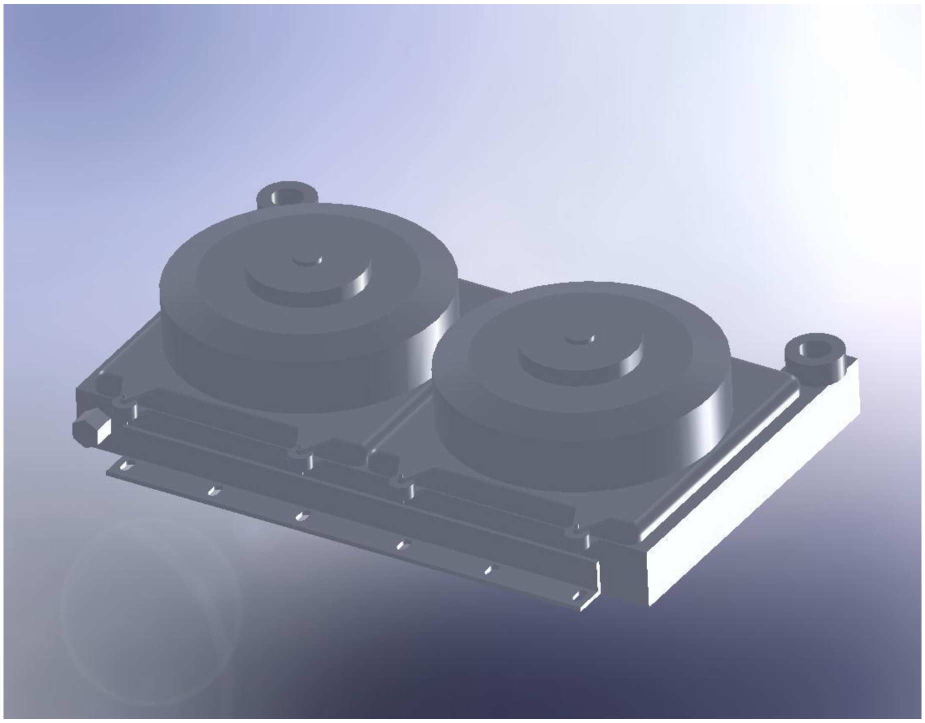


Figure 70: Cooling System Auxiliary Oil Cooler

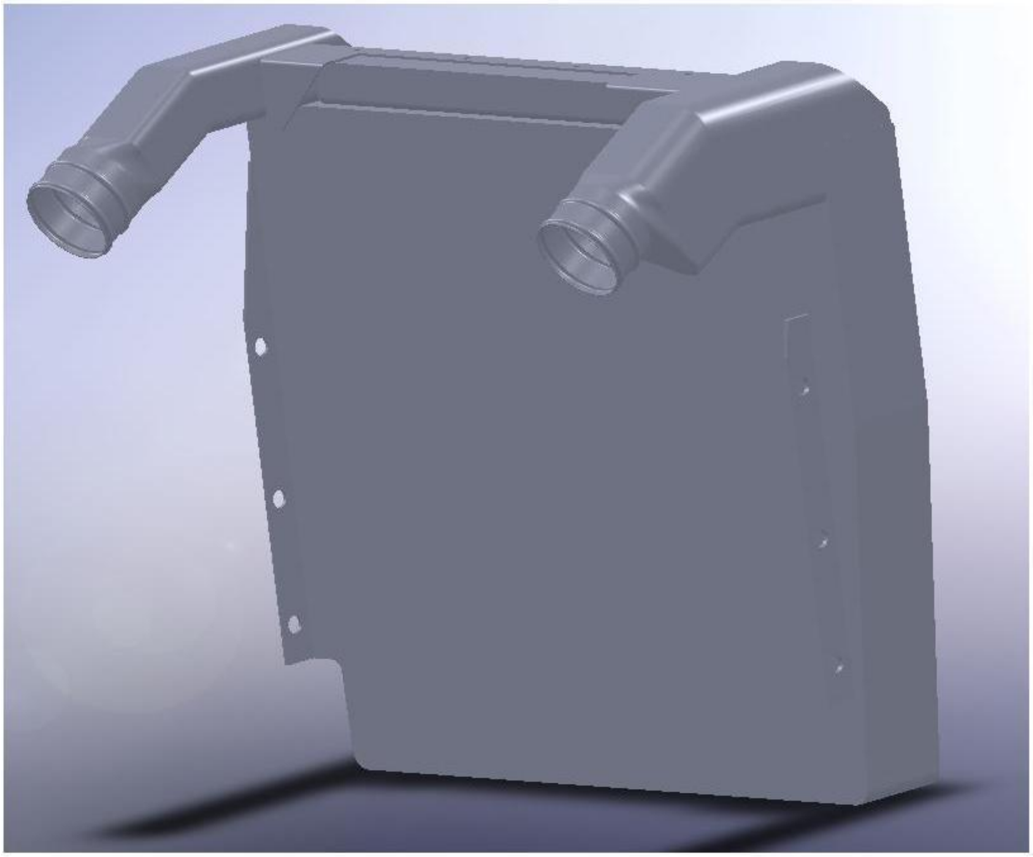


Figure 71: Cooling System Charge Air Cooler

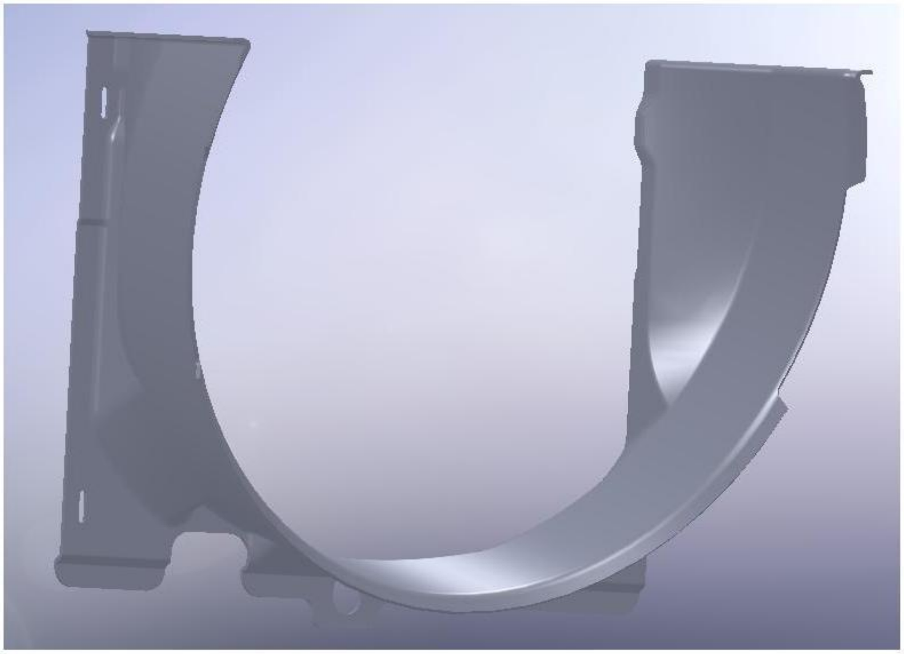


Figure 72: Cooling System Bottom Fan Shroud

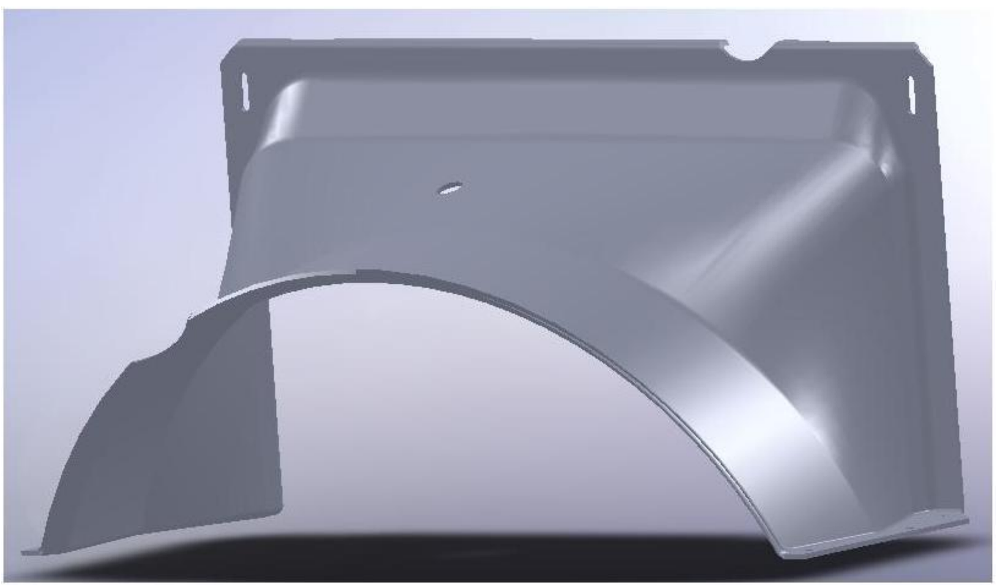


Figure 73: Cooling System Top Fan Shroud

Figure 74: Cooling System Cooling Fan

Figure 75: Cooling System Fan Clutch Component

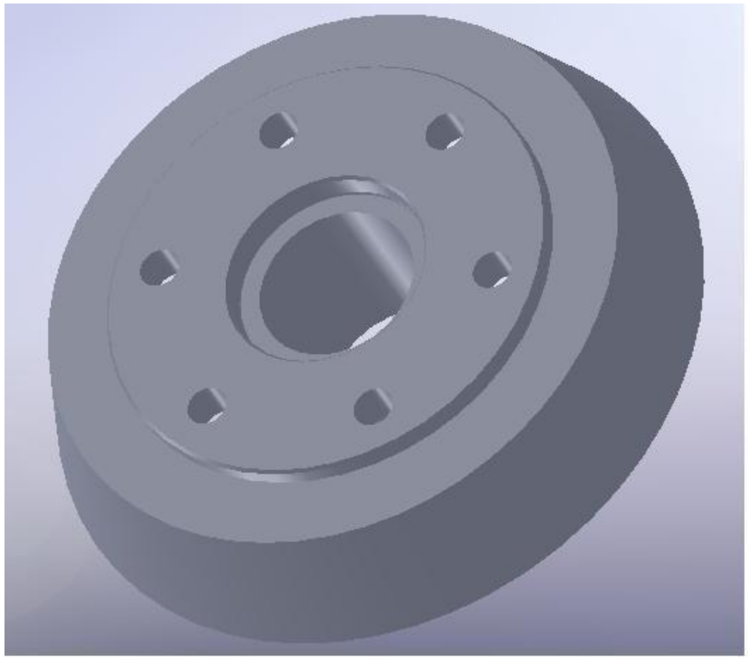


Figure 76: Cooling System Fan Clutch Component

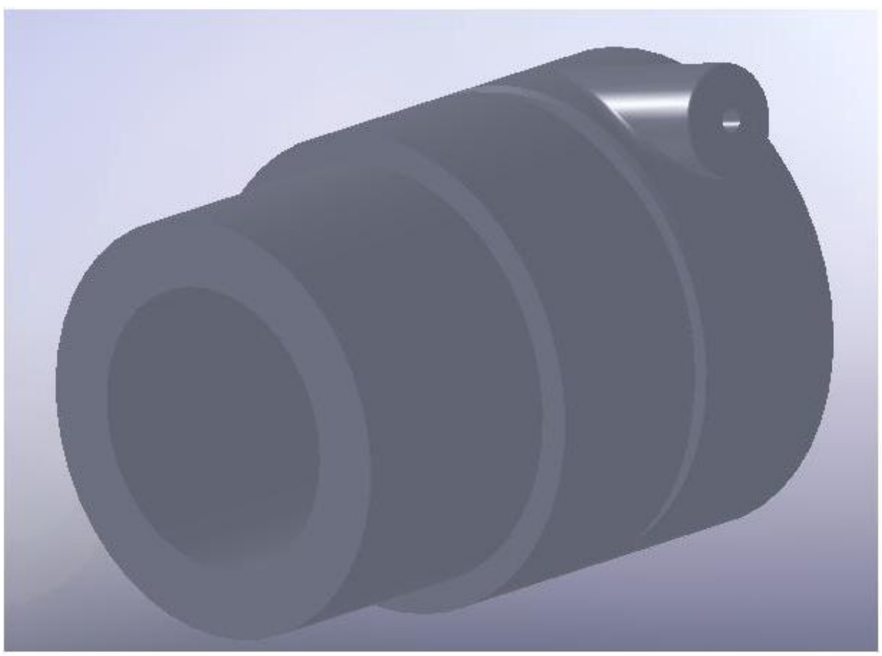


Figure 77: Cooling System Fan Clutch Component

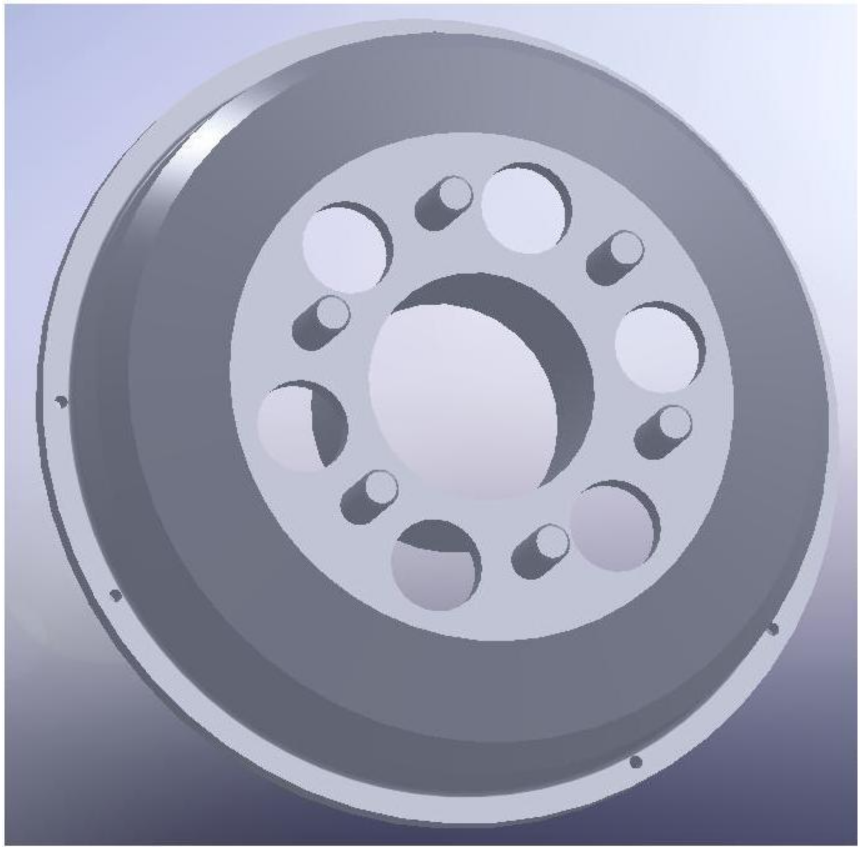


Figure 78: Cooling System Fan Clutch Component

FMTV Chassis Subsystem

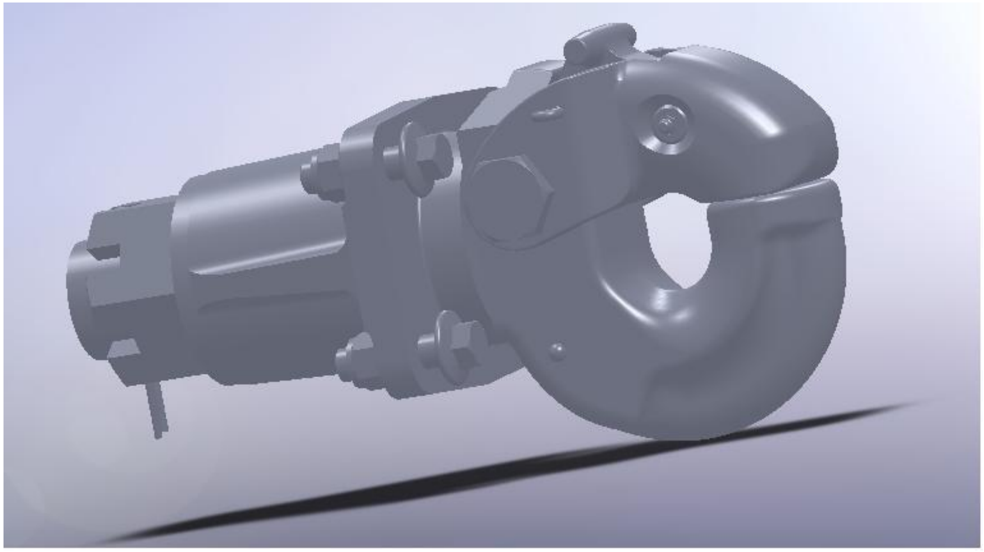


Figure 79: Chassis Subsystem Trailer Hitch

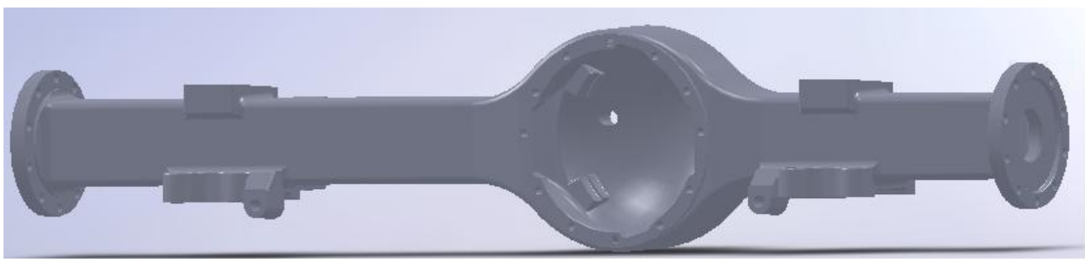


Figure 80: Chassis Subsystem Rear Axle Housing

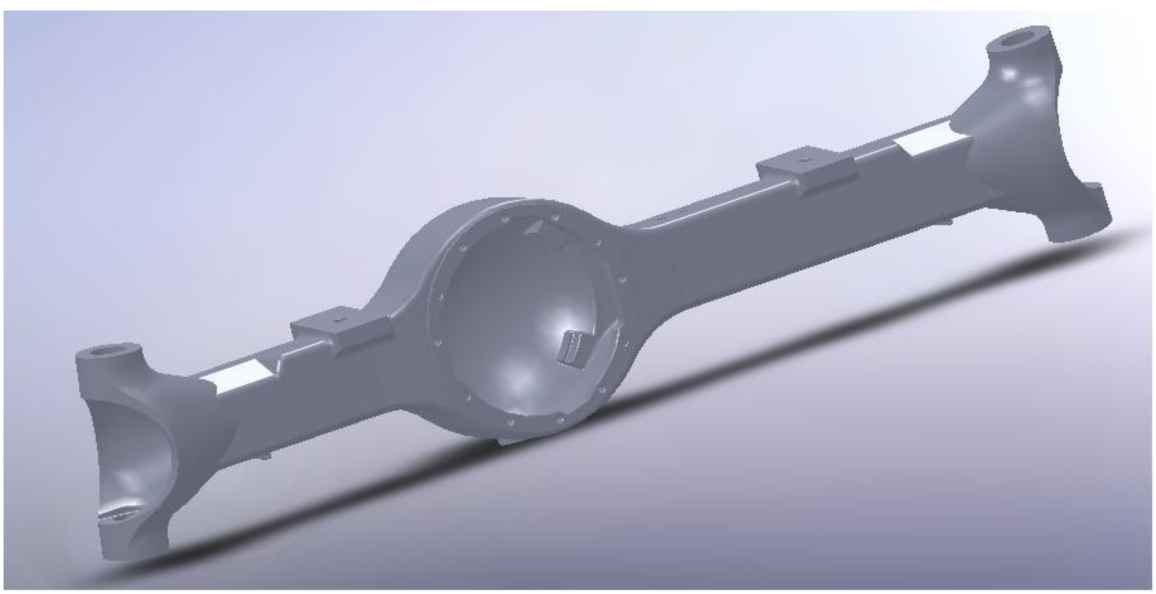


Figure 81: Chassis Subsystem Front Axle Housing

Figure 82: Chassis Subsystem Main Beams

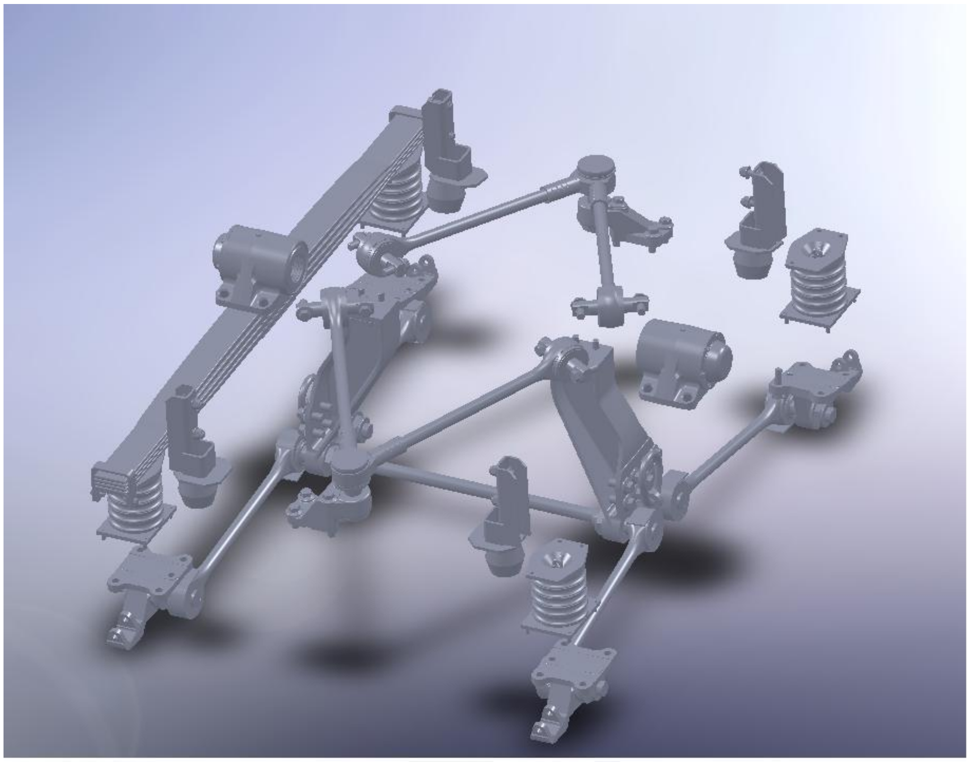


Figure 83: Chassis Subsystem Leaf Springs

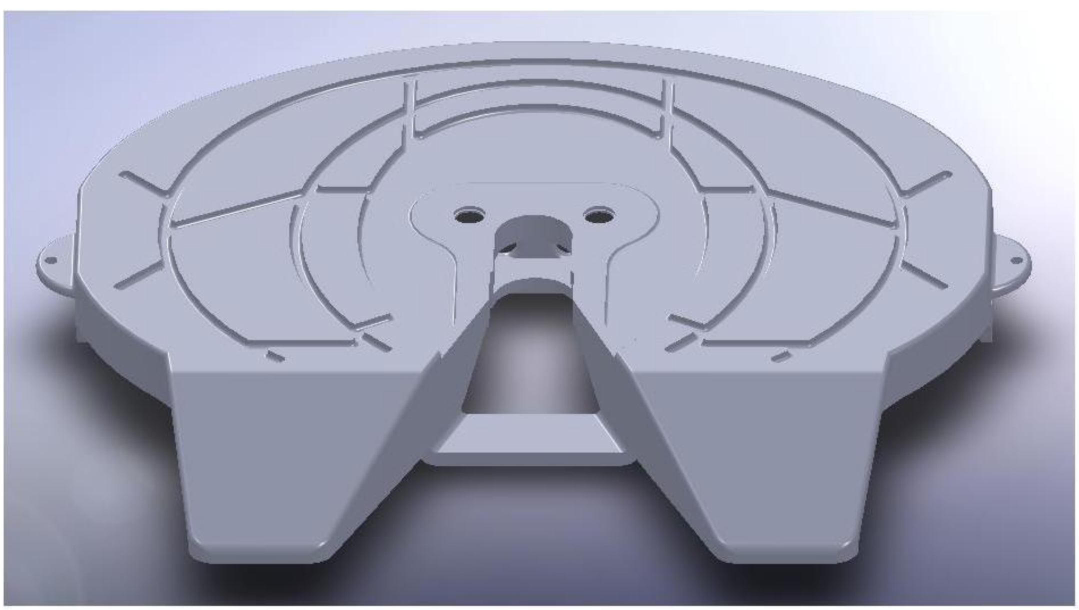


Figure 84: Chassis Subsystem Fifth Wheel

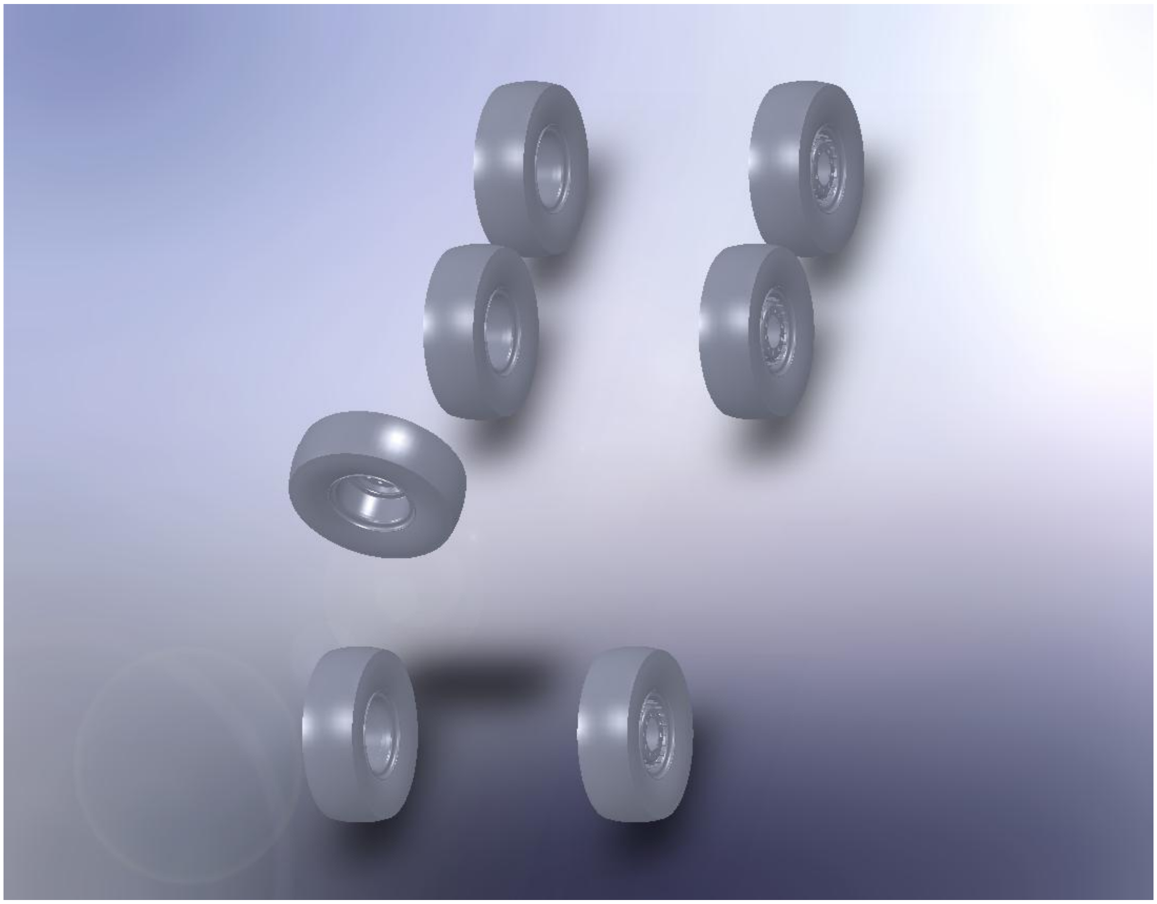


Figure 85: Chassis Subsystem Tires

FMTV Cab Subsystem

Figure 86: Cab Subsystem Steering Wheel

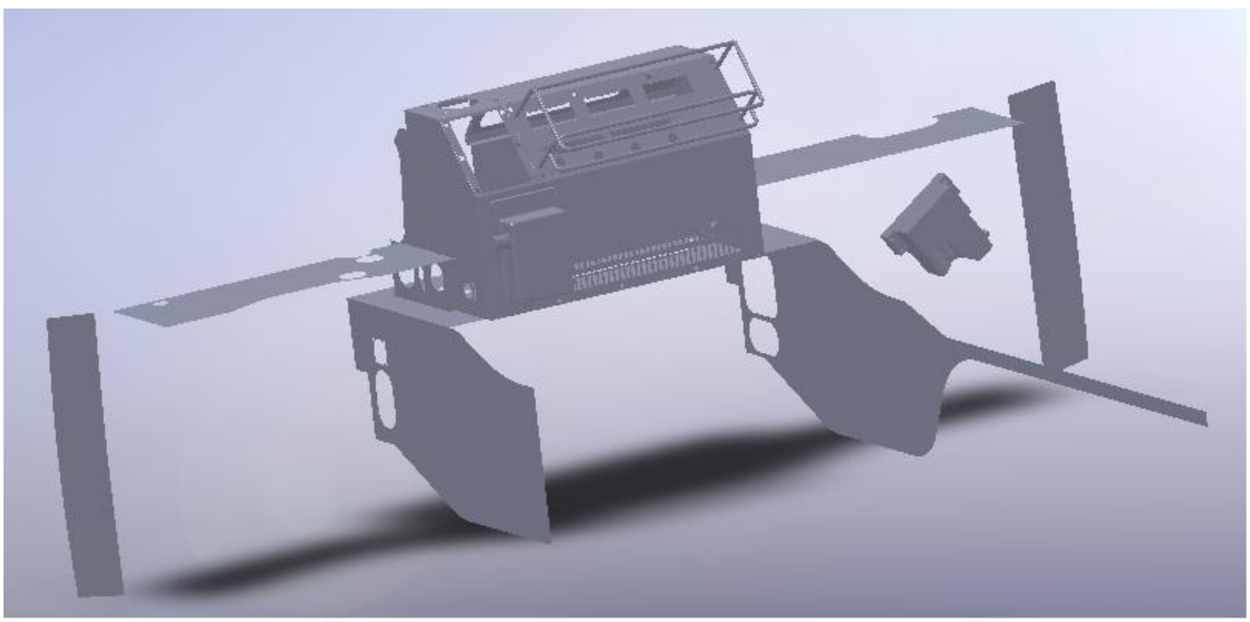


Figure 87: FMTV Cab Instrument Panel (1)

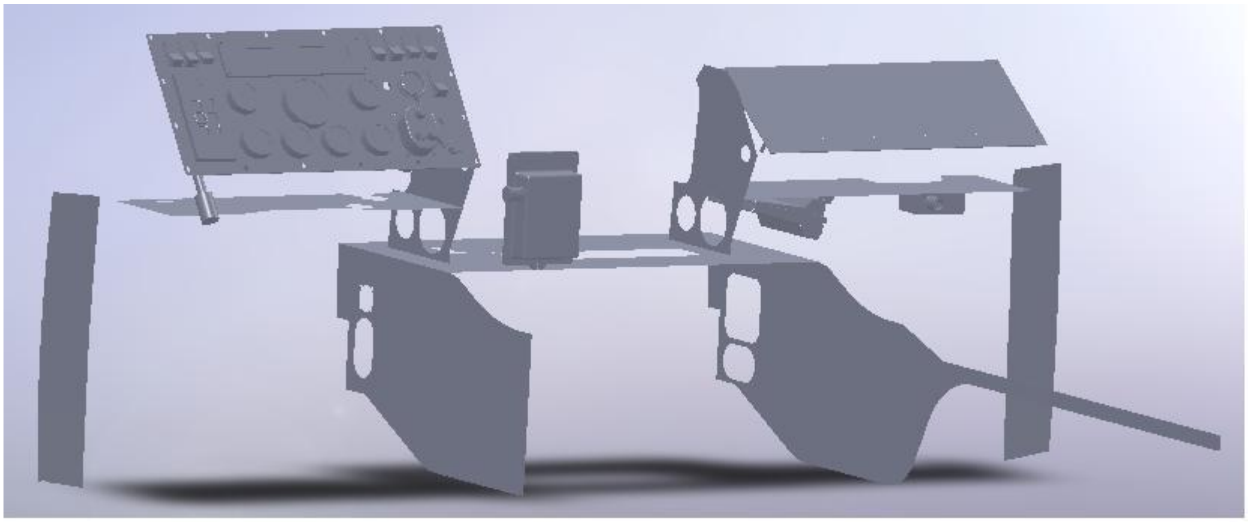


Figure 88: FMTV Cab Instrument Panel (2)

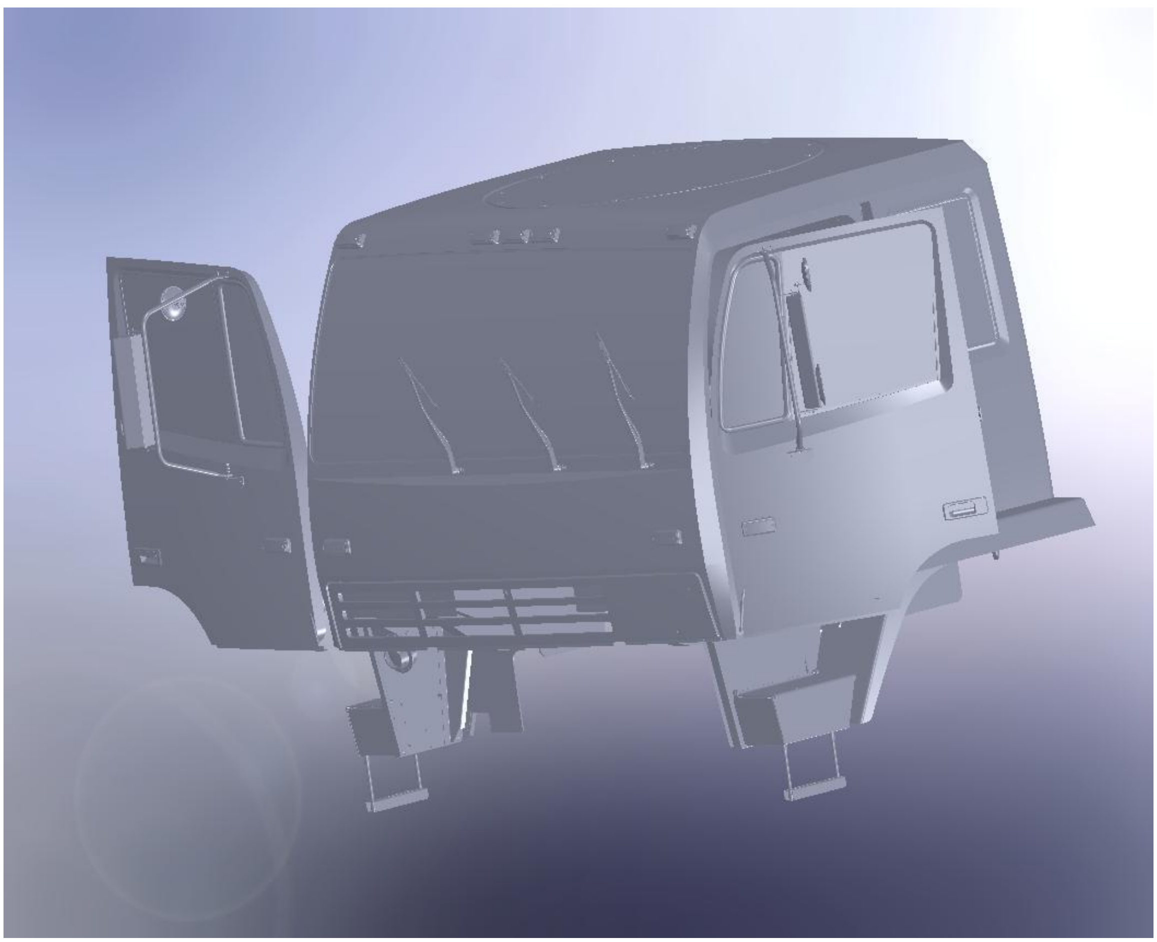


Figure 89: FMTV Cab Subsystem Housing

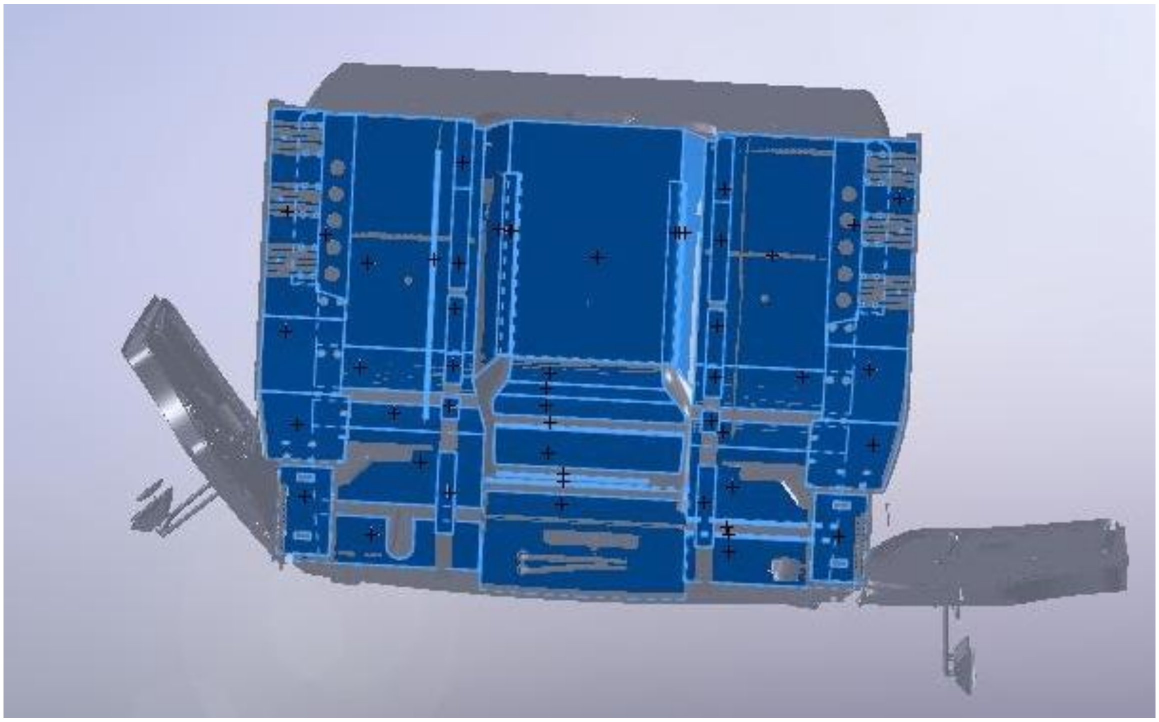


Figure 90: FMTV Cab Subsystem Housing