A Data Element Mapping and Analysis (DEMA) Approach for Implementing a Complete Digital Thread

by

Allison B. Ledford

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 5, 2023

Keywords: Data Element Mapping and Analysis (DEMA), Digital Thread, Digital Engineering, Model-Based Enterprise, Digitalization, Functional Analysis

Approved by

Gregory Harris, Chair, Professor of Industrial and Systems Engineering
Gregory Purdy, Assistant Professor of Industrial and Systems Engineering
Erin Garcia, Lecturer of Industrial and Systems Engineering
Gerry Dozier, Charles D. McCrary Eminent Chair Professor of Computer Science and Software Engineering
Phillip Farrington, Professor Emeritus of Industrial and Systems Engineering, The University of Alabama in Huntsville

Abstract

In the age of digitalization and the fourth industrial revolution, the concept of the Digital Thread and the Model-Based Enterprise has captured both the attention and resources of countless establishments. However, despite the growing popularity of digitalization and a plethora of technologies being offered that promise to create the Digital Thread and the Model-Based Enterprise, their full realization remains elusive. In this research, a novel approach for the standardized capture, mapping, and analysis of data flows for digital system understanding and architecture is developed and presented. Research projects were conducted with both a prototyping organization and a verification and validation operational environment to analyze the current state of data and information flows and use that as a basis for making improvements.

These efforts resulted in the creation, development, and maturation of Data Element Mapping and Analysis (DEMA)- a technology agnostic approach that allows an enterprise to move from a functional, document-centric, hierarchical view of data and information flows to a data element level view, with the data elements serving as the connectors of the Digital Thread. Once data elements are identified and categorized, they can be logically reorganized with the integration of digital technologies to create the connected Digital Thread and/or the Model- Based Enterprise in a way that reduces organizational risk, lead-time, and manpower while increasing traceability, quality, and profitability.

Acknowledgments

Firstly, I would like to thank my advisor, Dr. Greg Harris for his guidance and support throughout my graduate program. I would have never considered pursing this degree and finding my passion if it were not for him. Thank you for everything, Dr. Harris. You changed my life, and I am forever grateful. I would also like to thank my committee members for their support and contributions to this research. Dr. Gregory Purdy, Dr. Erin Garcia, Dr. Gerry Dozier, and Dr. Phillip Farrington- thank you so much for sharing your invaluable time and expertise. It has been a pleasure working with you.

I would like to thank each of the organizations that enabled me to conduct my research applications for my second contribution. I would also like to thank Dr. Ashley Yarbrough for her mentorship and friendship throughout my program. Ashley, your research inspired me to create DEMA, and I am so thankful.

Lastly, I would like to thank my family for their unwavering support. Mom and Dad, I would not be here today if it were not for you. I would like to thank my incredible husband, Jackson, for his unwavering support, patience, and help with coding syntax. Thank you to my loyal companion, Rufus, for being the best rescue dog and home office 'pal'. To my son Arthur- delivering you and completing this dissertation within a month of each other may have not been the most efficient planning on my part, but I would do it again and again for you. Mostly importantly, I would like to thank God for leading me down this path and sustaining me throughout this journey.

Table of Contents

Abstract ii

Acknowledgments iii

1 Introduction 1

1.1 Overview 1

1.2 Description of the Research Space 2

1.3 Theory Base for the Research 3

1.4 Aims and Objectives 4

1.5 Contribution of the Research 5

1.5.1 Development of the DEMA Approach 5

1.5.2 Application of DEMA Across Multiple Domains 6

1.5.3 Quantification of DEMA Metrics of Improvement 6

1.6 Outline of the Dissertation 7

2 Review of Literature 8

2.1 Introduction to Digitalization 8

2.2 Visual Mapping Techniques for System Analysis and Understanding 11

2.2.1 The History of Visual Mapping Techniques 11

2.2.2 Visual Mapping Techniques in the Digital Age 16

2.2.3 Previous Research in Analytical Techniques for Data and Information .. 18

2.3 The Digital Thread 20

2.4 The Model-Based Enterprise 22

2.5 The Digital Thread, the Model-Based Enterprise, and Industry 23

2.6 Metrics for Quantifying the Effects of Digitalization 25

3 Research Statement 27

3.1 Identification of the Research Gap 27

3.2 Problem Description 28

4 Methodology 31

4.1 Addressing Research Question 1 31

4.2 Addressing Research Question 2 32

4.3 Addressing Research Question 3 33

5 Development of Data Element Mapping and Analysis (DEMA) Approach 34

5.1 DEMA Step 1 39

5.2 DEMA Step 2 40

5.3 DEMA Step 3 44

5.4 Results from DEMA Development 48

6 Application of DEMA Across Multiple Domains 49

6.1 Results from DEMA Application in Product Realization Environment 49

6.1.1 Step 1: Functional Level Mapping and Analysis 51

6.1.2 Step 2: Data Vessel Level Mapping and Analysis 56

6.1.3 Step 3: Data Element Level Mapping and Analysis 64

6.2 Results from DEMA Application in a Modeling and Simulation Operational Environment 66

6.2.1 Step 1: Functional Level Mapping and Analysis 69

6.2.2 Step 2: Data Vessel Level Mapping and Analysis 72

6.2.3 Step 3: Data Element Level Mapping and Analysis 76

6.2.4 Identification of Disconnects in System Terminology and Knowledge …. 77

6.2.5 Comparison of Applications ……………………………………………79

7 Quantification of DEMA Metrics of Improvement.. 81

7.1 Evaluation of Metrics of Improvement.. 82

7.2 A Continuous Improvement Methodology for Digitalization.................................. 83

7.3 Application of Continuous Improvement Methodology for Digital Engineering..... 89

8 Conclusion and Future Work.. 108

8.1 DEMA and Systems Engineering... 109

8.2 DEMA and the Principles of Lean Manufacturing 111

8.3 DEMA and Digital Engineering 113

8.4 DEMA and MBSE.. 113

8.5 DEMA and A New Age for V&V and System Development............................... 115

8.6 Future Work.. 116

References 119

Appendix ……………………………………………………………………………134

List of Figures

2.1 Example of 1947 Operation Process Chart from ASME Standard [49]........ 14

2.2 Functional Flow Block Diagram Format Example [54]........ 15

2.3 Versions of IDEF from Center for Complex Systems & Enterprises [53] 17

2.4 Business Process Model and Notation Choreography Diagram [63]........ 18

2.5 Business Process Model and Notation Implementation Diagram [63] 18

5.1 IDEF0 Example from IEEE [118] 35

5.2 DEMA Overview........ 38

5.3 Steps to Create Functional Level Mapping........ 40

5.4 Data Vessel Mapping Syntax........ 43

5.5 Generic Data Vessel Mapping For Single Activity........ 43

5.6 Generic Example of Data Element Identification 45

5.7 Data Element Excel Sheet 47

5.8 Data Thread Isolated On Data Element Excel Sheet........47

6.1 High Level Functional Areas of the Additive Manufacturing Project 51

6.2 Sub-Functional Areas of the Additive Manufacturing Project........ 52

6.3 FBD of the Functional Activities of the Additive Manufacturing Project 53

6.4 FFBD of the Functional Activities of the Additive Manufacturing Project........ 54

6.5 Updated FFBD of the Functional Activities of the Additive Manufacturing Project. 55

6.6 Data Vessel Questionnaire for Application 1........ 56

6.7 Data Vessel Mapping of First Engineering Activity........ 57

6.8 Section of Color-Coded Engineering Data Vessel Mapping........ 59

6.9 Legend for Color-Coded Data Vessel Mappings 59

6.10 Program Management Color-Coded Data Vessel Mapping (Part 1) 60

6.11 Program Management Color-Coded Data Vessel Mapping (Part 2) 60

6.12 Engineering Color-Coded Data Vessel Mapping (Part 1) 61

6.13 Engineering Color-Coded Data Vessel Mapping (Part 2) 61

6.14 Verification & Validation Color-Coded Data Vessel Mapping.................................. 62

6.15 Fabrication Color-Coded Data Vessel Mapping 62

6.16 Quality Color-Coded Data Vessel Mapping... 63

6.17 Final Delivery Color-Coded Data Vessel Mapping.. 63

6.18 Data Element Mapping Syntax 64

6.19 Current State of an Isolated Data Element Thread of AM Part Nomenclature 65

6.20 Improved State of an Isolated Data Element Thread of AM Part Nomenclature 65

6.21 High-Level Functional Areas of the V&S Process of the Modeling and Simulation (M&S) Environment... 70

6.22 Sub-Functional Areas of the V&S Process of the Modeling and Simulation (M&S) Environment 71

6.23 Functional Activities of the V&S Process of the Modeling and Simulation (M&S) Environment.. 73

6.24 Data Vessel Mapping Activity 1.113 75

7.1 Roh et al's Proposed Metrics for Data and Information Flows [76] (Figure of Table from [113] 84

7.2 Yarbrough's Metrics for Data and Information Waste: Right Data (Figure of Table from [113]).. 85

7.3 Yarbrough's Metrics for Data and Information Waste: Right Place (Figure of Table from [113] 85

7.4 Yarbrough's Metrics for Data and Information Waste: Right Time (Figure of Table from [113] 86

7.5 Yarbrough's Metrics for Data and Information Waste: Right Form (Figure of Table from [113] 86

7.6 IDEF0 of a Continuous Improvement Methodology for Digital Engineering 87

7.7 A0: Apply DEMA to Current State System 90

7.8 A1: Determine Opportunities for Improvement 90

7.9 Pseudocode for Step A1 Code ……………………………………………………..91

7.10 High-Level Code Flowchart 92

7.11 Example of Current State Data Element Excel Sheet for A1 93

7.12 Example of Graph Structure From Rows in Figure 7.10 94

7.13 Example of Waste Flagged in Graph Structure From Rows in Figure 7.10………… 94

7.14 A2: Determine Improved State of Data Threads 103

7.15 A3: Verify Improved State of Data Threads 104

7.16 Data Vessel Architecture for First Function of Improved Engineering Data Flows ……105

7.17 Data Element Arch for First Function of Improved Eng, Data Flows 106

7.18 A4: Create Improved System Architecture ………………………………………..106

7.19 A5: Verify and Validate Improved System Architecture 107

List of Tables

5.1 DEMA Terminology……………………………………………………………34

5.2 Evaluation of Past Visual Mapping Techniques for Data Element Thread Isolation 36

5.3 Definition of Column Data in Data Element Level View Excel Sheet......................46

6.1 Definition of Column Data in Data Element Level View Excel Sheet (Second Application)……………………………………………………………..………78

7.1 Calculation of Labor Hour Metrics for Data Elements in Engineering Functions …93

7.2 Determination of Data and Information Wastes in Data Element Flows 96

7.3 Part 1: Current State Metrics for Data Elements Calculated by Code 97

7.4 Part 2: Current State Metrics for Data Elements Calculated by Code 98

7.5 Part 1: Prototyping Application Current State Metrics for Data Elements Calculated by Code 99

7.6 Part 2: Prototyping Application Current State Metrics for Data Elements Calculated by Code 100

7.7 Part 1: Prototyping Application Improved State Metrics for Data Elements Calculated by Code 101

7.8 Part 2: Prototyping Application Improved State Metrics for Data Elements Calculated by Code 103

Chapter 1

Introduction

1.1 Overview

It is commonly recognized that manufacturing is undergoing a fourth industrial revolution centered upon the integration of cyber-physical systems [1]. It has also been recognized that technology concepts within Industry 4.0., such as digitalization, have major implications beyond manufacturing for both business and society [2][3]. Although the full impact of digital transformation is still unfolding [3], it is estimated that digital transformation will create value in the trillions of dollars [4][5].

The advancements enabled by digital technologies also go beyond monetary value to provide significant enhancement in national security [6][7][8]. Therefore, there is a resolve across many industries to enable and realize the advances brought on by the fourth industrial revolution and digital transformation [3]. However, despite growing recognition of the importance of digitalization, there is a lack of consensus on a common lexicon [9][10][11]. To obtain the benefits enabled by the fourth industrial revolution and the age of digital transformation, it is crucial that key terminology be defined so that progress can be made.

One of the results of this lack of consensus in lexicon is the misuse of terms. For example, the terms "digitization" and "digitalization" are often confused with one another [7]. Whereas digitization is the computerization of manual activities, digitalization is the fundamental restructuring of an existing process to improve connectivity and information flows while taking advantage of digital capabilities [7]. While digitization is a perquisite for digitalization, digitization by itself does not result in a more efficient, secure, and advanced process.

Therefore, despite popular belief, digital transformation cannot be achieved by simply digitizing information or implementing digital technologies [7][3]. Digitalization requires a fundamental rethinking of the way a process functions. In this research, a generalizable method that synthesizes both existing and new systems analysis and elicitation techniques was created to identify and visually map data and information flows as a means of enabling the Digital Thread and the Model-Based Enterprise. Data Element Mapping and Analysis (DEMA) is a novel approach for the standardized capture, mapping, and analysis of data flows for digital system understanding and architecture.

1.2 Description of the Research Space

The work of incredible minds such as Frederick Taylor, Frank Gilbreth, and Dr. Lilian Gilbreth revolutionized the industrial world and laid the foundations for the discipline of Industrial Engineering [12] a century ago during the second industrial revolution. Manufacturing is now undergoing a fourth industrial revolution centered upon the integration of cyber-physical systems [1], and it will have major implications for both manufacturing and the world [2][3]. However, there is a lack of consensus in defining the key terminology surrounding digitalization [9][10][11]. With the excitement of digital technologies such as new cyber-physical systems, machine connectivity, and new software and hardware, it is often forgotten that data and information flows are at the heart of all digitalization.

Despite this confusion, there is a growing understanding of the need for new processes and tools beyond physical technologies to enable digitalization [6][7] with recognition of the importance of understanding data and information flows [13][14][15]. The field of Industrial Engineering has evolved significantly in the past few decades to include various aspects of other sciences including operations research, management science, computer science, and information systems [12]. As the field of Industrial Engineering has greatly broadened to become more quantitatively based [12], it is not surprising that it is often coupled with Systems Engineering.

Systems Engineering has been fundamental to the success of various institutions such as the National Aeronautics and Space Administration (NASA), who defines Systems Engineering as “a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system” [16]. A hierarchy of elements exists within almost any system [17][16]. Man-made systems have now reached complexities previously un- heard, and data is a key system element in addition to humans, software, hardware, and others [18].

Within the modern age, there is recognition that systems engineering philosophy is essential to the realization of the digitalization efforts [19][20][21][22]. Processes and procedures are key elements of a system [16]. Therefore, Industrial and Systems Engineering together are uniquely suited in today's modern age to enable the transformation of digital processes to reap the benefits of the fourth industrial revolution. This is the general space into which this dissertation research contributes.

1.3 Theory Base for the Research

At its most basic level, this research is founded upon the principles of Scientific Management introduced by Frederick Taylor with the idea that an optimum method can be identified and achieved to reach a certain goal [23]. This research is also founded upon the concept that systems can be visually mapped, a concept that for physical structures has been around for thousands of years [24] and for process flow since at least the time of the Gilbreths [23] around the time of the second industrial revolution.

The Toyota Production System, or as it is often called Lean Manufacturing, has also greatly influenced this work with the philosophy of the elimination of non-value added activities, or waste, within a system [25]. Another pillar on which this research stands is systems science, from which systems engineering draws heavily [26]. Within systems engineering, this research utilizes the theory of system decomposition, systematic elicitation techniques, and functional analysis. Parts of this research are also built upon the idea that graph theory can be applied to represent and link data flows [27].

In summary, this research is founded upon the science of process and the application of mathematical principles that can be used to decompose, analyze, and improve complex systems, specifically, within the area of data and information flows of digital systems. It is a desired outcome of this research that ideas that traditionally may have been compartmentalized into the categories of “Industrial Engineering” or “Systems Engineering” will be united by the ideas and contributions presented in this dissertation.

1.4 Aims and Objectives

This research has several aims in contributing to the body of knowledge. One significant desire of this work is to provide novel contributions that are of use to academia, industry, and government in reaching the potential of the digital age. Secondly, this research was conducted to bring attention to novel concepts within the field of digitalization, or more specifically, to the untapped opportunities within the understanding and improving data and information flows.

The objectives that have guided this research are outlined below:

1. Objective 1: To achieve and present a rigorous and thorough academic understanding of the body of literature surrounding the field of analytical tools that enable digitalization with identification of the research gaps in this field.

2. Objective 2: To develop and provide a novel analytical solution that serves to fill the research gap identified in Objective 1.

3. Objective 3: To present the analytical solution developed to meet the Objective 2 to various partners in industry and to apply them across various domains in a way that is beneficial to all parties to prove the generality, practicality, and usefulness of the tool.

4. Objective 4: To examine the results of the application of said tool across various domains and to develop quantifiable metrics of that confirm the efficacy of the tool and serve to contribute to the business case for digitalization.

5. Objective 5: To provide a path forward for future research and development that builds upon the contributions developed and presented in this work.

1.5 Contribution of the Research

This research was conducted to contribute to the field of digitalization by the development, application, and quantification of a methodology to aid in the systematic analysis and improvement of data and information flows. The tool described herein is named the Data Element Mapping and Analysis (DEMA) process. DEMA is applied across multiple domains to enable the realization of both the Digital Thread and the Model-Based Enterprise, and the benefits of implementation are quantified.

1.5.1 Development of the DEMA Approach

Data and information flows are at the heart of all digitalization, and therefore analytic tools are needed for the design, analysis, and improvement of systems that utilize digital technologies. A significant research gap exists as no suitable tools are available to accomplish this in a way that systematically uncovers and isolates threads of data to a data element level. The Digital Thread cannot be developed unless the relationships between individual data elements across the lifecycle are uncovered, captured, and understood. Therefore, this dissertation presents a novel approach for the standardized capture, mapping, and analysis of data flows at the data element view for digital system understanding and architecture.

The Data Element Mapping and Analysis (DEMA) approach was developed out of necessity while analyzing the data and information flows of a prototyping organization and while reviewing the body of knowledge that is presented in Chapter 2. DEMA was created using system decomposition methods, elicitation techniques, and the principles of functional analysis. The development of DEMA serves to fulfill Objective 2 of this research by providing a systematic tool for enabling the realization of Industry 4.0 and digital transformation concepts such as the Digital Thread and the Model-Based Enterprise. The DEMA approach is currently patent pending with various options for commercialization being considered.

1.5.2 Application of DEMA Across Multiple Domains

Objective number three of this research is to present the analytical solution developed in various industry applications and to apply it across various domains to demonstrate the generalizability, practicality, and usefulness of the tool. The second contribution of this research is the application of the DEMA tool across multiple domains. This is accomplished through two diverse applications of DEMA. A significant amount of interest has been shown by industry for further development and application of the DEMA approach.

The first application of DEMA is in a product realization, prototyping, environment. DEMA was created and developed while attempting to analyze the data and information flows of the organization from prototype ideation to delivery to the customer using known analysis techniques. In this application, DEMA is specifically applied with the aim of enabling the Digital Thread. The second application of DEMA is in a verification and validation operational environment for modeling and simulation. DEMA was applied to this environment as a means of enabling the Model-Based Enterprise in addition to the Digital Thread. Whereas the first application of DEMA was to an existing and defined prototyping process, the second application enabled the development of a non-existent verification and validation operational process. Both applications included the initial stages of the organizations implementing new data and information flows based on the outcomes and analysis of the DEMA approach. With each of these applications, the results are presented with analysis and metrics in Chapter 5.

1.5.3 Quantification of DEMA Metrics of Improvement

Metrics are needed to prove the benefits of digitalization and to further prove the efficacy of the Data Element Mapping and Analysis approach. Whereas the previous work in this dissertation provided the basis and proof for the theory and practical applications of DEMA, this part of the research seeks to prove the benefits of DEMA in a quantifiable manner using the result from the applications. This part of the dissertation work involves the development and utilization of a continuous improvement methodology for digital engineering and architecture.

1.6 Outline of the Dissertation

This dissertation is structured according to the following outline, with high-level descriptions for each chapter.

1. Introduction: This chapter introduces the work as a whole and provides the context of the research.

2. Review of Literature: This portion of the work presents the detailed review of literature that served as the foundations for each contribution.

3. Problem Description: In this chapter the research gap identified from the literature is identified, and the problem description is presented.

4. Methodology: The method by which this research is conducted is explained in terms of the development of DEMA and the methods by which DEMA was applied and validated.

5. Development of Data Element Mapping and Analysis (DEMA): The results of the first contribution are presented.

6. Application of DEMA Across Multiple Domains: The results of the second contribution are presented.

7. Quantification of DEMA Metrics of Improvement: The results of the third contribution are presented.

8. Conclusion and Future Work: Concluding thoughts are presented and the path forward is outlined.

Chapter 2

Review of Literature

This review of literature seeks to achieve and present a rigorous and thorough academic understanding of the body of knowledge surrounding the field of analytical tools that enable digitalization. The research overviewed in this chapter sets the stage for the identification of the research gaps in the field.

2.1 Introduction to Digitalization

Manufacturing is undergoing the fourth industrial revolution that is centered upon the integration of cyber-physical systems [1]. Within Industry 4.0., various concepts such as digitalization have developed, with implications beyond manufacturing for both business and society [2][3]. The full impact of digital transformation is still unfolding [3], but it is estimated that digital transformation will create value in the trillions of dollars [4][5]. The implementation of the Digital Thread for interoperability, seamless transmission, and advanced analytics of data is valued as a $30 billion opportunity annually within manufacturing alone [27].

The enrichment acquired by implementing digitalization extends beyond monetary value to provide significant enhancement in national security [6][7][8]. The United States Department of Defense has recognized Digital Engineering (digitalization) as necessary to achieve greater performance, improve response time, reduce costs, and increase vendor engagement with the United States military [6]. The DoD has also identified digital modernization as an essential strategy to enable mission success [8]. The recognition of the importance of the fourth industrial revolution and digitalization go far beyond just the United States to include many governments [2][28]. The German government in particular is well known for the identification of the fourth industrial revolution, or “Industrie 4.0” and taking a very proactive approach to enable it with national policy [2].

Therefore, there is an opportunity across all industries and academia to enable and realize the advances brought on by the fourth industrial revolution and digital transformation.

According to the consulting group McKinsey, two thirds of industrial companies across the world report that digitizing their operations as one of their most important priorities [29]. One area that has captured growing interest is that of enabling implementation of advanced manufacturing technologies in small and medium-sized enterprises [28][7]. Another is that of establishing public-private partnerships between industry, academia, and government to enable advanced manufacturing [30][31].

Manufacturing is experiencing an unprecedented and "explosive" rise in capabilities that enable Digital Manufacturing [31]. Despite growing recognition of the importance of advanced manufacturing and digitalization, there is a lack of consensus in defining its key terminology [9][10][11]. Two key terms that are often confused are "digitization" and "digitalization" [7]. Whereas digitization is the computerization of manual activities, digitalization is the fundamental restructuring of an existing process to improve connectivity and information flows while taking advantage of digital capabilities [7]. While digitization is a perquisite for digitalization, digitization by itself does not result in a more efficient, secure, and advanced process. Some in research have gone so far as to distinguish digitization, digitalization, and digital transformation as the three stages that collectively make up digital transformation [32].

Despite popular belief, digital transformation cannot be achieved by simply digitizing information or implementing digital technologies [7][3] and digital transformation does not occur without specific organizational structures being enacted [32]. Common misconceptions such as these must be overcome, and various researchers are striving to define key terminology with the digital revolution [7] [9] [2].

Within the fourth industrial revolution and digitalization, the concepts of the Digital Thread and the Model-Based Enterprise have captivated both academia and industry. The Digital Thread is defined as the connection of data and information flows throughout a system lifecycle [20][19][33][21].

The Digital Thread enables the capture of lifecycle data and allows for its use and maturation at the appropriate time by the users of the system [31]. Utilization of the Digital Thread through efficient data and information transfer is valued as a $30 billion annual opportunity for the manufacturing industry [27]. The Digital Thread is often confused with the concept of the Digital Twin [34]. Whereas the Digital Thread is the integrated connection of system lifecycle data, the Digital Twin is model-based and consists of a physical system, a virtual system, and bidirectional data flow between them [35][34].

Even with this distinction, there is still confusion as to what a Digital Twin actually entails and how it is different from modeling and simulation [10]. Researchers are actively working to achieve clarity for the Digital Twin so that it can successfully be understood and implemented [36][37]. Another popular concept is the Model-Based Enterprise. The Model-Based Enterprise (MBE) is an organization that uses modeling and simulation to manage and integrate all lifecycle data [38]. Whereas the Digital Thread is focused on the connection of lifecycle data, the Model-Based Enterprise specifically makes meaningful connections in lifecycle data through the development and application of models.

Therefore, the Model-Based Enterprise may use models as a as means of connecting the data to create the Digital Thread, but they are two different concepts. Within the Model-Based Enterprise, Model-Based Systems Engineering, or MBSE, has grown in popularity across various industries [39]. It is worth noting that just as in the implementation of digital technologies, the use of models by an organization does not mean that the Model-Based Enterprise and its many benefits, have been achieved. With all of this confusion and excitement, it is not surprising that there are barriers that must be overcome to enable the benefits of digital transformation and Industry 4.0.

There is a strong need for multidisciplinary research to enable digital transformation [32]. In research conducted by McKinsey, more than 700 persons from over 50 countries across various industries were interviewed, and the results revealed that many organizations are struggling to move from advanced manufacturing pilot projects forward to affecting their bottom line [29]. In benchmarking studies conducted by KPMG, similar results were found with most organizations remaining in a low-to-medium stage of maturity in the implementation of various Industry 4.0 technologies [40]. Overall, the review of general Industry 4.0 and digital transformation literature indicates that although many feel digital transformation is necessary, progress is slow, and a great deal of confusion exists in defining and implementing key technologies that will provide significant benefits.

There is also a recognition that man-made systems have now reached complexities previously unachieved, and data is a key system element in addition to humans, software, and hardware [18]. It is well understood and established that a hierarchy of elements exists within almost any system [17]. Therefore, the review of literature also indicates that there is a recognition that systems engineering philosophies are essential to the realization of digitalization concepts such as the Digital Thread [19][20][21][22].

Processes and procedures are key elements of a system [16]. Therefore, Industrial Engineering and Systems Engineering together are uniquely suited in enabling the transformation of digital processes that are necessary to reap the benefits of the fourth industrial revolution. Visual mapping techniques used in Industrial Engineering and Systems Engineering to enable system design, analysis, and improvement can be used to enable digital transformation.

2.2 Visual Mapping Techniques for System Analysis and Understanding

Both Industrial Engineering and Systems Engineering have originated various visual mapping techniques for system analysis and understanding. Visual mapping techniques in these fields have traditionally been utilized for system design, analysis, and improvement. This section shows how the traditionally separated principles of Industrial Engineering and Systems Engineering analytical tools can be combined to enable the transformation of digital processes to reap the benefits of the fourth industrial revolution.

2.2.1 The History of Visual Mapping Techniques

The practice of using visual mappings for system understanding and analysis has been utilized for thousands of years. Some of the earliest drawings for technical purposes were created by the Egyptians in 2500 B.C. [24]. Some of the earliest drawings for physical structures are from the time of Mesopotamia in 2200 B.C. [41]. The use of technical drawings advanced through the use of inventors such as da Vinci and Descartes [42].

The first known recorded geographic map was created by the Babylonians around 600 B.C. well before Ptolemy introduced some of the first applications of mathematics to maps in his textbook “Geography” around 150 A.D. [43]. Architectural drawings became widespread and accurate starting in the 13th century [41]. Visual maps were used primarily to describe physical systems until around the time of the 2nd Industrial Revolution when there was significant advancement in the field of engineering drawings. The first electrical-engineering curriculum was developed and implemented at MIT in 1882 [44], and Alfredo Bensaude established drawing as an official discipline in the engineering curriculum at the Instituto Superior Te´cnico (IST) [45].

It was at this time of increasing industrialization and advancement in engineering standards that there was a movement away from a strictly mechanical and electrical mindset of cause-and-effect and tools for process analysis and improvement were developed to meet the needs of the second industrial revolution. In 1881, Frederick W. Taylor introduced the concept of time study and lay the foundation for his work as the father of scientific management [46].

Process science began to merge with visual mapping techniques, and this transformed the way industry designed and viewed manufacturing processes. Soon after, Henry Ford established the Ford Motor Company and revolutionized manufacturing with mass production techniques [23]. However, the second industrial revolution was not the first time that process science was deployed for designing and managing complex systems.

Some of the most influential work on process concept evolution had occurred thousands of years prior with military strategies recorded in *The Art of War* [23]. Other process evolution occurred hundreds of years prior with Adam Smith's observations on work processes which later inspired Taylor's *Scientific Management* [23]. Therefore, the process concept evolution driven by military strategy and Adam Smith's observations laid the foundation for Frederick Taylor to develop the scientific management principles that would later inspire visual mapping techniques for process.

As Taylor conducted time studies, his contemporaries Frank and Dr. Lillian Gilbreth developed motions studies as a scientific means of identifying and eliminating waste in the physical motions of workers [47], and Frank Gilbreth developed the "process chart" and the "flow diagram" of which American Society of Mechanical Engineers (ASME) later standardized [25]. A copy of an example operation process chart taken from ASME's 1947 *Operation and Flow Process Chart* developed by the ASME Special Committee on Standardization of Therbligs, Process Charts, and Their Symbols [48] is shown in Figure 2.1.

This tool quickly became popular within engineering curriculum, and in the 1940's businesses such as Proctor and Gamble capitalized upon it to develop their Deliberate Methods Change Program [49]. After thousands of years of the drawing of physical systems, Gilbreth had created a tool that captured process. Thus began a new era in the age of visual mappings. Advancements in visual mappings then continued at a rapid pace. Henry Gantt, a disciple of Frederick Taylor and contemporary of the Gilbreths, advanced the technical problem-solving tools of scientific management by the creation of the "Gantt Chart" [50].

The Gantt chart is a type of bar chart that keeps track of a project schedule by illustrating progress using the start and finish dates of the elements of a project and filling in progress between those points [25]. Modern Gantt Charts show relational dependencies between project elements [25]. The Gantt Chart is now the most popular chart used in project management and is often created using Microsoft and other software [51]. In the 1930's, Allan Mogensen further developed Frank Gilbreth's process charts into more detailed process diagrams [25].

William E. Boeing employed visual process flow and breakdown practices to aid in the manufacturing of aircrafts produced for use in the second World War, and Ben S. Graham applied process practices to simplify paperwork and business activities in offices environments [25]. With this evolution occurring in the way of engineering processes, it is not surprising that other innovations in systems analysis were occurring by various professionals in the development of quality control, corporate operations management, and operations research [52].

In the 1950's, the intention embodied in the process chart evolved into several functional visual mapping techniques such as Functional Flow Block Diagrams (FFBD), Data Flow Diagrams (DFD), and Integrated Definition (IDEF) [25]. The purpose of the Functional Flow Block Diagrams is to describe a system's requirements in lifecycle functions and to define elements of the system [53]. Functional Flow Block Diagrams enable functional decomposition and are often created with hierarchical views of the system [53]. An example of the Functional Flow Block Diagram taken from the 2001 Department of Defense System Management College *Systems Engineering Fundamentals* [53] is shown in Figure 2.2.

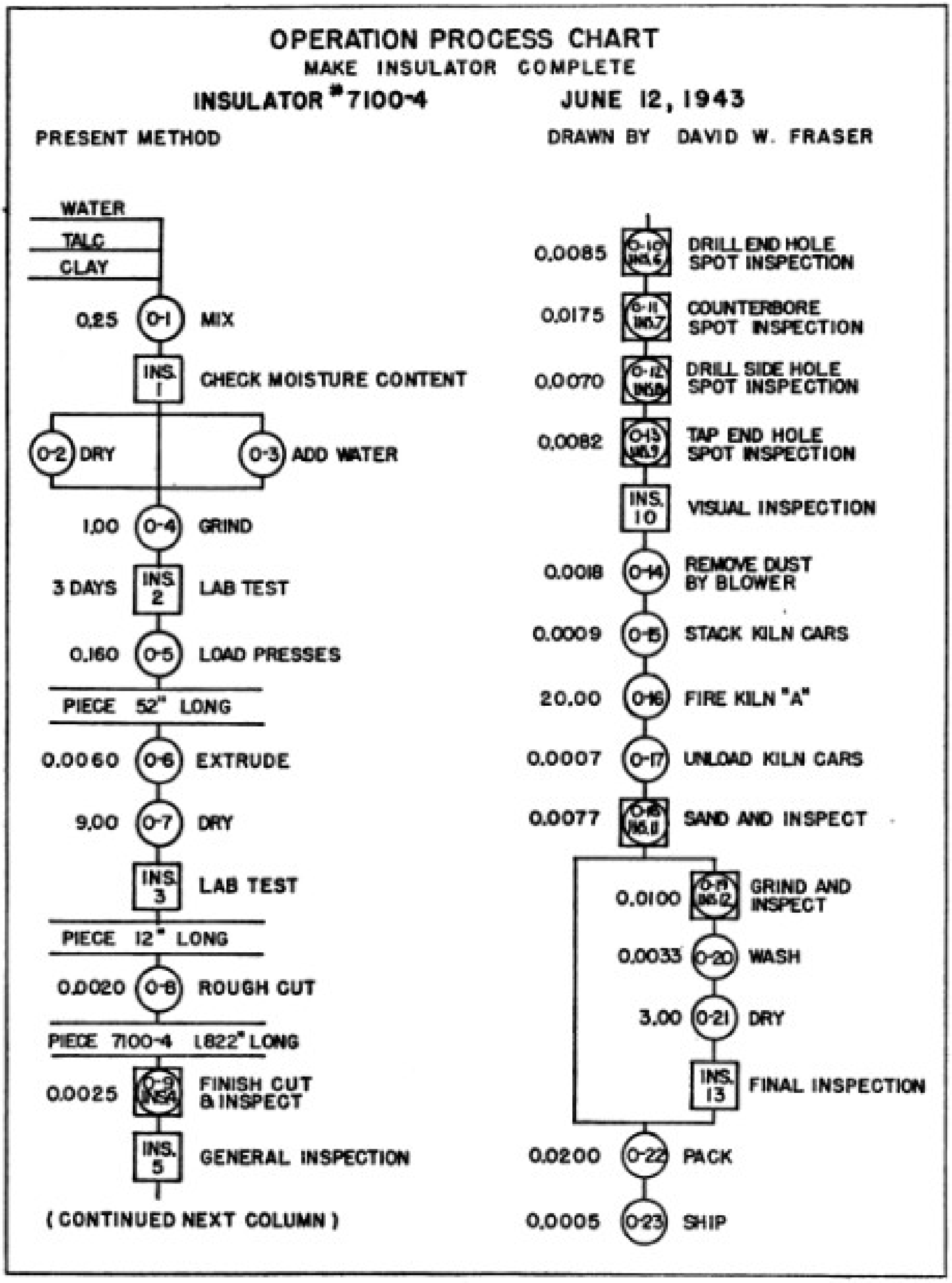


Figure 2.1: Example of 1947 Operation Process Chart from ASME Standard [49]

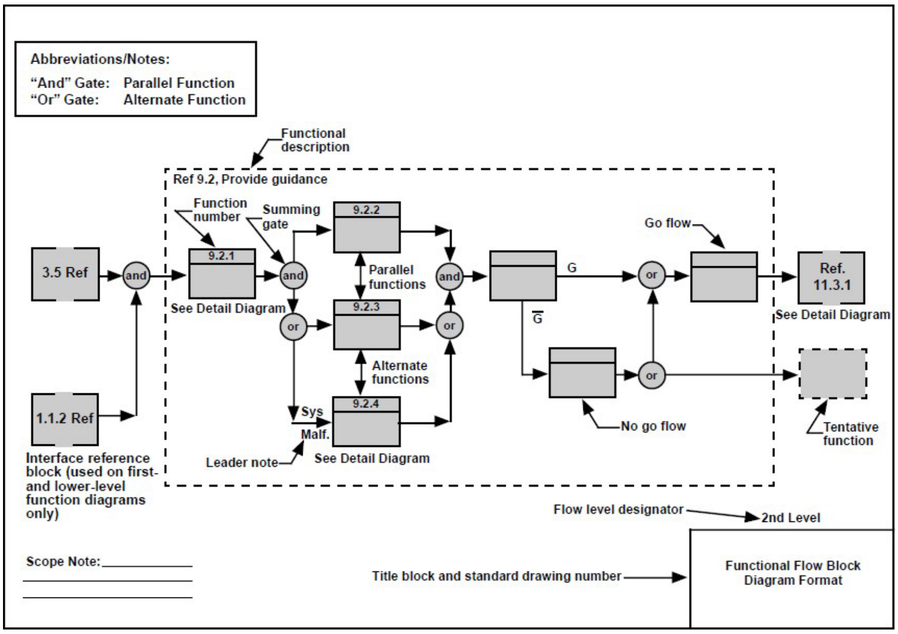


Figure 2.2: Functional Flow Block Diagram Format Example [54]

Data Flow Diagrams are created by placing information blocks into the flow of a process to show how and where data is stored and to provide information on how inputs are delivered and in what order [25]. Integration Definition for Function Modeling (IDEF) was introduced by Knowledge Based Systems, Inc. (KBSI) and introduced various methods of function modeling [25]. Control Flow diagrams rose in popularity in the 1950's [54]. The Toyota Production System was also being developed around this time [55], and within Toyota, information and material flow diagrams were created and were later made popular by the 1998 Lean Enterprise Institute book *Learning to See* as "value stream mapping" [56].

One of the concepts Taiichi Ohno developed under the Toyota Production System was that of "just in time" (JIT) manufacturing with the Kanban inventory control system that uses visual cues such as cards to prompt actions needed to drive a process [57]. Although Kanban itself is not necessarily a visual mapping, tools such as Kanban boards have been developed to visually map out its action tracking information [58]. Another functional visual mapping tool that was later developed was the N^2 diagram. This tool was developed to examine and understand system interfaces by placing system function blocks in the diagonal square of a matrix and then filling the remainder of the squares with the interface inputs and outputs between each function [16].

2.2.2 Visual Mapping Techniques in the Digital Age

Visual mapping techniques continued to evolve as the computer introduced a revolution in communicating, capturing, and managing information. The need arose for analytical tools that could also be used to better design, understand, and improve digital processes. In the modern age, the concept of the Control Flow Diagram is often used to determine the order in which a statement will be executed in a program, and Data Flow Diagrams are used in software engineering to visualize the flow of information that a user will experience [54].

Although Functional Flow Block Diagrams are still often used for software development and business processes [59], in 1993, the National Institute of Standards and Technologies (NIST) released standards for the Integration Definition for Function Modeling (IDEF0) [60]. IDEF0 differs from Functional Flow Block Diagrams in that it allows for visualization of the flow of data and other controls in addition to functional flow [53]. IDEF0 was based on a tool created by the Air Force [60].

The IDEF version known as IDEF0 is a very well-known and tested modeling technique for analysis, development, re-engineering, and integration of information systems, and is used as well in business processes and software development [53]. IDEF0 is commonly used in modern digital applications. Many versions of IDEF have since been introduced for various purposes and to address other aspects of a system [25]. Each version lies under the authority of different specifications with no integrated modeling framework to combine them [52]. The versions of IDEF and their purposes are shown in Figure 2.3.

In the 1980's the attention to the relationship between process and quality became a major focus in improvement approaches, with concepts such as Six Sigma and DMAIC gaining attention [55]. The business world was also soon affected even more powerfully by tools such as the architecture of integrated information systems (ARIS) and business process management software that helped couple information technology with process modeling [55]. In the 1990s, Michael Hammer's management theory of Business Process Reengineering (BPR) gained popularity and was reportedly adopted by 80% of Fortune 500 companies [55]. A scientific view of process had gained popularity even more broadly in the world of manufacturing and engineering.

Generation	Purpose
IDEF0	Function modeling
IDEF1	Information modeling
IDEF1X	Data modeling
IDEF2	Simulation modeling design
IDEF3	Process specification capture
IDEF4	Object-oriented design
IDEF5	Ontology description capture
IDEF6	Design rational capture
IDEF7	Information system auditing
IDEF8	User interface modeling
IDEF9	Business constraint discovery
IDEF10	Implementation architecture modeling
IDEF11	Information artifact modeling
IDEF12	Organization modeling
IDEF13	Three schema mapping design
IDEF14	Network design

Figure 2.3: Versions of IDEF from Center for Complex Systems & Enterprises [53]

In the early 2000s, Business Process Model and Notation (BPMN) was created to ensure standardization of various business process modeling techniques and to enable such models to be executable [61]. Business Process Model and Notation is a standard for creating visual mapping for Business Process Diagrams. There are various types of Business Process Diagrams with a number of different symbols to represent varying business activities. An example of a BPM Choreography Diagram is shown in Figure 2.4 and an Implementation Diagram is shown in Figure 2.5 [61]. However, despite the growing knowledge and development of new business process tools, the reality was that efforts often failed from a lack of organizational cultural transformation [62].

It is also worth noting that there were three significant periods in the evolution of data exchange standards that show a progression from developments such as Standard for Exchange of Product data (STEP), Extensible Markup Language (XML), and Unified Modeling Language (UML) to the development of system modeling language, SysML, and ontology-based data standards [63]. This progression towards modeling language demonstrates a transition towards the integration of data standards with more user-friendly modeling languages that have aspects of visual mapping techniques.

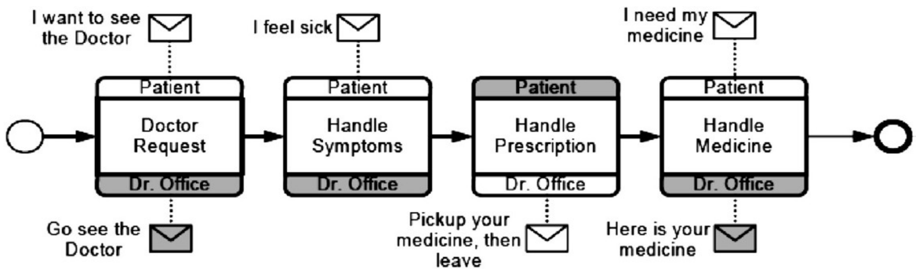


Figure 2.4: Business Process Model and Notation Choreography Diagram [63]

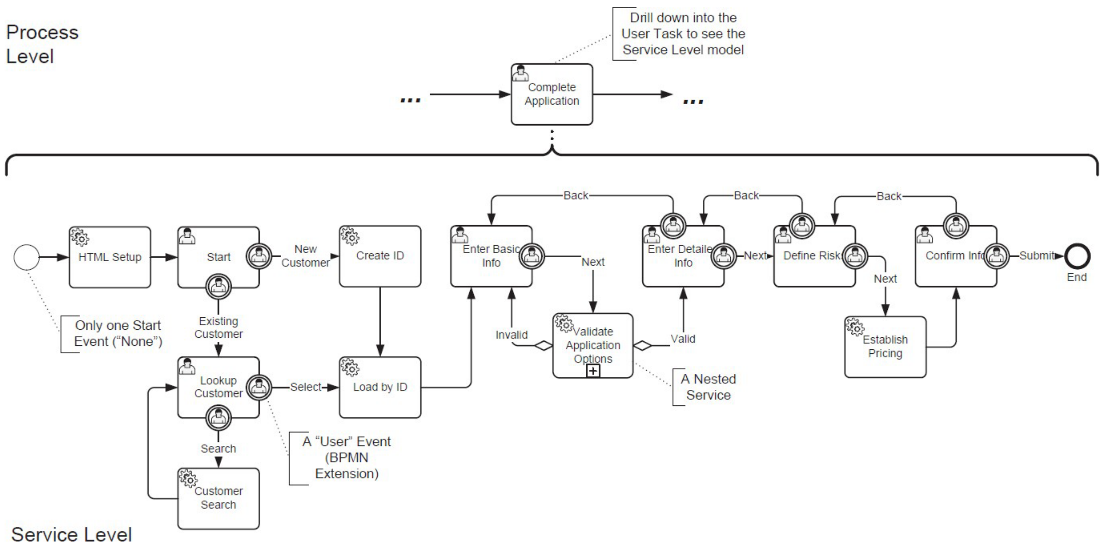


Figure 2.5: Business Process Model and Notation Implementation Diagram [63]

2.2.3 Previous Research in Analytical Techniques for Data and Information

Many of the visual mappings discussed in previous sections were established to meet the needs of the previous three industrial revolutions. Alfredo Bensaude's engineering drawing curriculum, the International Electrotechnical Commission (IEC)'s standard for electric circuit diagrams, Frank Gilbreth's Process Flow Chart, and Henry Gantt's "Gantt Chart" were created around the time of the Second Industrial Revolution. Data Flow Diagrams (DFD), the origins of IDEF, and N^2 diagrams were established at the time of the Third Industrial Revolution. In the same way visual mappings were created to meet the needs of previous industrial revolutions, novel mappings techniques are necessary to successfully implement and reap the benefits of the Fourth Industrial Revolution, and some research has been conducted to enable this capability. There is a growing understanding of the need for new processes and tools beyond physical technologies to enable digitalization [6][7] and recognition of the importance of understanding data and information flows [13][14][15]. While the total cost of inefficient data and information flows may be hard to fully quantify, there is no question that anecdotal data exposes its immensely negative impact. Within the United States Department of Defense acquisition process, obtaining test data from test ranges regularly takes 60 days because of disparate data repositories, manual data searches, and un-standardized data formats [22]. Also, the variety of digital tools used in engineering processes result in significant gaps in design flows that are bridged by ad-hoc, manual user interventions that result in variety of costs including labor, risk, and loss of opportunity [64], and the same difficulty and costliness in managing heterogeneous data and information holds true in manufacturing [65][66][14]. It is usually taken for granted that such manual and costly data tasks are necessary, and consequently, little thought is given to improvement [15].

Various work has been conducted in utilizing visual mapping techniques such as those presented in the previous subsections in analyzing data and information flows. Systemigrams are visual mappings that represent complex systems using natural language, and they have been used for decades in various settings [67]. More recently they have been used to create conceptual views of the DoD acquisition enterprise as a means of enabling the Digital Thread [22][68]. IDEF0 diagrams have been used to describe data flows in the Digital Thread in the work of [69] and [70]. Firesmith [71][72] has proposed a method for creating “End-to-End” Data Flow Diagrams by utilizing interview techniques and system documentation to identify and functionally map key data actions and components in a system for mission thread analysis. N^2 diagrams can be used to identify interfaces and analyze data interfaces [73].

There is also a recognition that a systems engineering philosophy is essential to the realization of the Digital Thread [19][20][21][22]. This makes sense given that the decomposition of complex systems into smaller groups is very useful in managing complexity and promoting system understanding [15]. Other research has tackled the issue of confusion between the roles of data architecture and information architecture and recognizes that no concrete solutions have been proposed to clear these misconceptions [74].

One of the most novel concepts in this area is that Lean Manufacturing tools and approaches are being used to eliminate waste in data and information flows. Work has even been performed in developing Taiichi Ohno's mental model for waste identification to be used in non-traditional applications such as to identifying waste in data and information flows [15]. However, it has been found that the 7 physical wastes that Ohno identified in manufacturing settings do not translate into data and information system analysis [15]. Other researchers are working to identify what these digital wastes may be [75]. Research has even been conducted in mapping information flows in ways inspired by value stream mapping to eliminate waste in manufacturing information inefficiencies [76][77]. It cannot be taken for granted that a system's current state of capturing, storing, and utilizing data is the most optimal and additional analysis is warranted.

2.3 The Digital Thread

The Digital Thread is a concept within digital transformation that has captivated countless organizations [21]. The term Digital Thread was reportedly coined by Lockheed Martin and the United States Airforce during the development of the F-35 [78]. The Digital Thread is defined as the connection of data and information flows throughout a product lifecycle [20][19][33][21] [78]. Like many concepts within digitalization there is confusion surrounding the concept of the Digital Thread. This is not surprising, as the name "Digital Thread" is actually a misnomer as it implies a single thread connecting all data.

Terms such as "Digital Tapestry" [79] or "Digital Quilt" [80] are more suitable names as their connotations more accurately convey the complex nature of interconnecting heterogeneous lifecycle data. Regardless of what the most appropriate name may be, the term "Digital Thread" appears to be the most popular title and is thus used in this dissertation. Although the implementation of digital technologies does not necessitate digital transformation [3], it is a key enabler of its realization, and several technologies have risen as popular choices in enabling it in the form of the Digital Thread.

Much research has been conducted into the technological solutions needed to enable the Digital Thread. Its popularity is not surprising, as research shows that enduring interest in Digital Thread is not unfounded. Within the manufacturing industry, connecting data and information flows are estimated to have annual savings of billions of dollars [33][81][21] and establishing the digital thread across supply chains should reduce cycle time by 75% [33]. Some research has shown that the monetary value that could be tapped into by the realization of the Digital Thread and advanced data utilization is at least $30 billion for the manufacturing industry [27].

The recognition that novel tools are needed to understand the data and information flows that make up the Digital Thread is becoming more well-known and has even been academically published [7]. Data collection and analysis techniques such as surveys [13], workshops [24], and comparison studies [82] have been conducted as an attempt to determine the necessary data flows to create Model-Based Definition and the Model-Based Enterprise.

There is also a recognition that a systems engineering focus is essential to the realization of the Digital Thread [19][20][21][22], as "adding digital capability to a bad process only creates a bad digital process" [7]. Also, the variety of digital tools used in engineering processes result in significant gaps in design flows that are bridged by ad-hoc, manual user interventions that result in variety of costs including labor, risk, and loss of opportunity [64]. The same difficulty and costliness in managing heterogeneous data and information holds true in manufacturing [65][66][14].

One concept that has arisen in Systems Engineering is that of the Mission Thread. The Mission Thread is a set of activities that must take place to accomplish a mission [83]. Whereas the Digital Thread is the connection of lifecycle data across a system, the Mission Thread is functionally focused on how a mission is accomplished. Data could be involved in the Mission Thread, but the Mission Thread goes beyond data to focus on activities. Approaches have been proposed to map the key data actions and items in Missions Threads using Data Flow Diagrams [71][72].

There are three significant periods in the evolution of data exchange standards that show a progression from developments such as Standard for Exchange of Product data (STEP), Extensible Markup Language (XML), and Unified Modeling Language (UML) to the development of system modeling language, SysML, and ontology-based data standards [63]. There are diverse efforts in employing data exchange developments such as these to enable data connectivity needed for Digital Thread realization. For example, a standards based approach has been proposed as an alternative method for linking data as opposed to costly and often siloed PLM (product lifecycle management) systems [84]. Within the field of software engineering, methods for model-based soft- ware synthesis [85][86] and model and tool integration platforms [64] have been put forward as solutions for integrating disparate systems. Other work has focused on means for integrating modeling languages [87] and creating novel data structures [69] in line with existing data integration standards as means of enabling the Digital Thread.

2.4 The Model-Based Enterprise

The Model-Based Enterprise is another concept that has developed alongside the era of digitalization. The definition of products is becoming exceptionally complex and 2D drawings are unsuitable for properly capturing them [24], and therefore Model-Based Definition (MBD) has been proposed as an alternative solution to meet the needs of the modern enterprise. In 2011, Frechette of the National Institute of Standards and Technology stated that the Model-Based Enterprise, or MBE, is "an organization that applies modeling and simulation technologies to integrate and manage all of its technical and business processes related to production, support, and product retirement" [38].

Like the Digital Thread, the Model-Based Enterprise has been identified as having key advantages such as a reduction in quality errors and lead time [88]. Also like the Digital Thread, there are misconceptions about what Model-Based Enterprise is. It is important to clarify that the word "model" does not necessarily imply a 3-dimensional geometric model but can also mean either a computational or descriptive model [88]. Within the Model-Based Enterprise, there are several sub-concepts. Model-Based Engineering is phrase used to describe model-based efforts specifically within the engineering domain of the enterprise [89].

The Model-Based Definition (MBD) specifically refers to a Digital Engineering artifact that defines the requirements and specification of the product and that can be used to perform the engineering functions across the Model-Based Enterprise [90]. Some have even described MBD to be beyond an artifact and to also encompass the strategy employed in its use across the lifecycle [13]. Regardless, it is clear the MBD must serve as one of the major points of connectivity if it is to be a component of the Digital Thread. Various standards have been proposed by organizations such as ASME and ISO to establish governance over what information is required to complete the MBD [13].

Work is being conducted in both the area of determining how a "complete" MBD is accomplished [13] and how to utilize the "minimum" information needed for transfer throughout the Model-Based Enterprise [91]. This indicates that like any complex problem within the Model-Based Enterprise, there are multiple objectives that must be meet for an optimal utilization of models and that at the heart of this optimization lies the efficient utilization of data. Within the Model-Based Enterprise is the concept of Model-Based Manufacturing and Inspection, which refers specifically to the utilization of model-based practices in manufacturing and inspection and of which there is recognition of the importance of Digital Thread connectivity [92].

2.5 The Digital Thread, the Model-Based Enterprise, and Industry

Although the Digital Thread and the Model-Based Enterprise can both be considered under the umbrella of digitalization, they are two distinct concepts. Whereas the Digital Thread is the connection of all lifecycle data for proper access, transmission, and storage by users, the Model-Based Enterprise specifically uses models as a means of managing data. Therefore, a Model-Based Enterprise may assist in enacting the Digital Thread by modeling data in a digital form that is suitable for connecting to the Digital Thread, but the MBE does not systematically connect all relevant lifecycle data.

Models are technically not required for the Digital Thread as other means of connectivity could be used such as a PLM system. However, the use of models offers significant advantages such as a reduction in quality errors and reduced lead time [88]. Therefore, implementing both the Digital Thread and the Model-Based Enterprise together within an organization has significant potential for unlocking value in data utilization.

Various types of software for the storage, transmission, and management of data have been presented to realize the Digital Thread. Product Lifecycle Management (PLM) software is used to manage information and processes across lifecycles [93][94][84] and is advertised as the "foundation" of the Digital Thread [95]. Even CAD systems have been modified to have some abilities to capture knowledge for reuse in the product lifecycle [24], and various cloud platforms have started to be adopted by the manufacturing industry [96].

Various technologies for capturing and re-using knowledge across product lifecycles have also been proposed [97][24]. In terms of cyber-security, blockchain technology has emerged as a popular means of securing data traceability in Digital Thread applications to ensure data authenticity [98]. It is worth noting that many works focusing on data exchange technologies for enabling the Digital Thread [69][98][99] are specifically within the Additive Manufacturing domain.

Within the Model-Based Enterprise, Model-Based Systems Engineering, or MBSE, has captivated diverse sectors including defense, commercial, and healthcare industries [39]. Delgatti has defined three pillars, or enablers, of MBSE as modeling languages, modeling methods, and modeling tools [100]. The three significant periods in the evolution of data exchange standards show a progression towards system modeling language, SysML, and ontology-based data standards [63]. This shows a progression towards user-friendly modeling tools for exchanging data that have aspects of visual mapping techniques.

Within the field of Model-Based Systems Engineering (MBSE), SysML has emerged as the "de facto standard" [101]. A system model in SysML defines and shows the interconnections between elements of the system that represent key aspects, and there are various types of diagrams within SysML to create system models [102]. Therefore, SysML is both a language for data exchanges and a visual mapping technique, further proving the interdisciplinary nature of digitalization efforts.

Efforts made toward Digital Thread and Model-Based Enterprise implementation are multi-disciplinary and active across diverse institutions. ASTM International has recognized that value stream mapping across industries is necessary to identify the key components of the Digital Thread for additive manufacturing [30]. There is significant need for public-private partnerships to enable digitalization efforts, and several institutions have been set up to meet digitalization needs such as the Digital Thread. In the United States, MxD (Manufacturing x Digital), formerly known as Digital Manufacturing and Design Innovation Institute (DMDII), was created to tackle problems such as those facing the implementation of the Digital Thread and the Model-Based Enterprise [31]. Another organization is CESMII, the United States non-profit institute dedicated to Smart Manufacturing [103].

The National Institute of Standards and Technology is well-known for its work in the

Digital Thread and Model-Based Enterprise space. Various societies and governing bodies have also taken interest in leading the way in defining standards and working groups to enable the Digital Thread and the Model-Based Enterprise. The American Society of Mechanical Engineers (ASME) formed the Model-Based Enterprise (MBE) Standards Committee to oversee the development of rules, guidance, and use cases related to the Model-Based Enterprise, and they have various working groups under its authority [104]. Overall, the review of literature shows that there is significant interest in the Digital Thread and the Model-Based Enterprise across diverse industries with various organizations attempting to implement it.

2.6 Metrics for Quantifying the Effects of Digitalization

With the introduction of any new concept, organizations require some sort of proof of benefit before spending time, money, labor, and other valuable resources to implement new concepts and tools. Digitalization concepts such as the Digital Thread and the Model-Based Enterprise have immense promise in the minimization of hidden costs and improved quality throughout a system. However, digital transformation is a disruptive concept, and implementing and managing disruptive technologies is a daunting task that leaves some organizations paralyzed with apprehension [3]. Therefore, key business cases and proof of ROI are key for organizations attempting a digital transformation.

Estimates showing the immense value of implementing digitalization and principles of the fourth industrial revolution are becoming more common and growing. As stated earlier, it is estimated that digital transformation will create value in the trillions of dollars [4][5]. Within the manufacturing industry, utilizing the Digital Thread for interoperability, seamless transmission, and advanced analytics of data is a $30 billion annual opportunity [27]. However, there are estimates that the cost of software development and sustainment for a Digital Thread of a large-scale Department of Defense program could be approximately $80 to $180 billion dollars [105]. Therefore, new tools are needed for implementing digitalization, and there must be quantitative proof that such tools are beneficial in terms of cost and value brought to the system.

Previous research has attempted to quantify the effects of digitalization. Waters and Ceruti presented a method for the modeling and simulation of information flow by using an agent-based model that treated information, suppliers, and consumers as agent particles [106]. Other work has used agent-based modeling to explore the effects of data flows on supervisors giving promotions to employees [107].

Work has even been performed in the field of modeling data and information flows across social media platforms. Pond [108] investigated how network structure impacts information flow and modeled an information theoretic approach to measure information flow across social platforms. More recently, work has been done in utilizing value stream mapping and simulation techniques to model data and information wastes in a manufacturing system [109]. Overall, the review of literature found that there is limited work in the field of quantitative modeling and optimization of the efficiency of data and information flows.

Chapter 3

Research Statement

3.1 Identification of the Research Gap

The diversity in the visual mapping techniques created over the past century demonstrates the many different functional views can be employed for system understanding and analysis. It also indicates that some are more useful than others for specific applications. Although many of the visual mapping techniques presented here can be used to visualize the flow of documents, their functional perspective inhibits the ability to isolate threads of data elements (the flow of individual units of data such as dimensions, document titles, and meeting times). This means that they are suitable for taking functional, document, and software-centric views of data and information, but the mapping objects used make it impossible to see where there are breaks in individual data threads.

Other methods have revealed and presented data and information flows in a way that is narrative in nature, and although narrative results can provide invaluable insights, they cannot provide detailed architectures of data and information flows. A technique is needed for exposing, mapping, and analyzing the complex and disconnected data and information flows found in real-life systems. Visual mapping techniques are a viable solution for this purpose. Also, none of the mapping techniques presented earlier offer a systematic method for exposing the complexities of the hidden data and information flows that exist in almost all systems. Therefore, previous visual mapping techniques are suited more for the manual computerization of activities, or digitization [7]) rather than true digitalization.

The review of the literature indicates that there is a significant gap in the research in the need for a visual mapping technique that moves beyond a functional and document view to achieve a data element level view. A visual mapping technique should be developed to meet this need. In terms of the Digital Thread and the Model-Based Enterprise, the review of literature also indicates that the efforts of academia and industry have been almost solely focused on the software, syntax, standards, and semantics necessary for data exchanges.

There is a common belief that the Digital Thread is solely the infrastructure in which data resides and is connected. While work that has been done to enable this infrastructure is essential to the realization of the Digital Thread, this research proposes infrastructure itself does not constitute or automatically create the Digital Thread. Therefore, a significant gap exists in providing analytical tools that can be used alongside infrastructure technologies that are essential to the realization of the Digital Thread and Model-Based Enterprise.

Traditional visual mapping techniques such as Business Process Model and Notation (BPMN), Value Stream Mapping, IDEF0, and Data Flow Diagrams can be used to visualize the flow documents, but their functional, document-centric, and hierarchical perspective inhibits the ability to isolate threads of data. The literature review found that the closest work to an approach for systematically uncovering data element threads was that of Donald Firesmith [71][72] with his End-to-End Data Flow method. An E2E Data Flow Diagram is made by utilizing interview techniques and system documentation to identify and functionally map key data actions and pieces in a system [71][72]. However, E2E Data Flow Diagrams take a functional and document level view instead of a data element level view.

3.2 Problem Description

An analytical approach that will lead the user through an iterative process of uncovering progressively complex views of the system and allow for the isolation of data element threads for enabling the Digital Thread and the Model-Based Enterprise (MBE) does not currently exist. While the utilization of technologies such as Product Lifecycle Management (PLM) software and Systems Modeling Language (SysML) software environments can potentially serve as mechanisms for connecting the Digital Thread and creating the Model-Based Enterprise, they themselves do not provide methods for systematically capturing hidden processes and data and information flows. Therefore, analytic tools are needed to better identify, understand, and quantify performance metrics for data and information flows.

The fact that a hierarchy of elements (related subsystems and components) exists within any system is a concept that is well understood and established [17]. It is recognized that man-made systems have now reached levels of complexities previously unachieved, and data is a key system element in addition to humans, software, hardware, and others [18]. Knowledge management has even been prescribed as a solution for the lack of mechanisms for integrating knowledge in smart manufacturing [97]. Knowledge management is a critical enabler of competitiveness in product realization environments [110]. It is worth noting that tribal knowledge includes both knowledge of products as well as the processes that create them [111].

In practice, organizations rely heavily on highly skilled employees utilizing siloed tribal knowledge to accomplish functional activities. Such systems are not sustainable as they rely on heroic effort from the user and are dependent on employees doing the right thing every time and remaining indefinitely within the organization, neither of which is feasible. Therefore, challenges such as these were crucial to consider in addressing the research gap. In complex systems, there will almost always be differences in how the different actors within the system conduct activities with different patterns of data exchanges within each activity. In these instances, the differences must be accounted for in any solution presented by this research.

As no such tool currently exists, there is also a significant gap in the research concerning the application and quantifiable proof of the efficacy of such a tool. Therefore, this research strives to contribute to the body of knowledge within the field of Industrial and Systems Engineering, and more specifically, the field of the Digital Thread and the Model-Based Enterprise by the development and application of an approach for the identification, visualization, communication, and analysis of the data elements that are fundamental to the existence of both. Also, the review of the literature indicated that there is limited knowledge within the field of modeling the efficiency of data and information flows in a quantitative manner, and therefore, this research will contribute to the body of knowledge in the identification of the effects of the application of an approach that solves the problem described.

The following research questions will guide the contributions of this research:

1. Can the analysis of data and information flows be achieved so that the fundamental unit of the data element is identified, documented, and visualized?

2. Could a method for the standardized capture, mapping, and analysis of data flows composed of data elements be utilized to realize the Digital Thread and the Model-Based Enterprise?

3. Can the efficiency of the data flows captured using such a method be quantitatively measured and systematically improved?

Chapter 4

Methodology

4.1 Addressing Research Question 1

The review of literature revealed that there is a significant research gap highlighting need for a visual mapping technique that moves beyond a functional and document view to allow for the isolation of individual threads of data elements. The review of literature also showed that no analytical approach exists that allows for the systematic capture of hidden processes and data and information flows while uncovering progressively complex views of the system down to the data element level view. The first step of the methodology for this research is to address Research Question 1: Can the analysis of data and information flows be achieved so that the fundamental unit of the data element is identified, documented, and visualized?

Research Question 1 is addressed by investigating the flows of data and information for a product realization process in a prototyping organization. Process Flow Charts, Integration Definition Methods, Control Flow Diagrams (CFD), Data Flow Diagrams (DFD), Functional Flow Diagrams (FFD), Functional Flow Block Diagrams (FFBD), Business Process Modeling Notation (BPMN), Value Stream Mapping (VSM), Unified Modeling Language (UML), Systems Modeling Language (SysML), N^2 diagrams, and E2E Data Flow Diagrams are evaluated for suitability in analyzing the organization's data and information flows.

Each of these visual mapping techniques is investigated to see if they can represent data storage and actor access. They are also evaluated based on whether they allow data element thread identification to be both possible and practical (easy to read). Given that organizations rely heavily on highly skilled employees utilizing siloed tribal knowledge to accomplish processes, each technique is evaluated for a systematic method for uncovering hidden processes, data, and information flows that are found in real-life systems.

Based on the results of the evaluation of past visual mapping techniques, an analysis tool will be developed that allows for the fundamental unit of the data element to be identified, documented, and visualized. This tool will integrate various traits of traditional functional mapping techniques, systems engineering elicitation methodologies, and novel data element thread isolation methods to move from a partial view to a holistic data element level view of a system.

4.2 Addressing Research Question 2

The review of literature revealed that a significant gap exists in providing analytical tools that can be used alongside infrastructure technologies that are essential to the realization of the Digital Thread and the Model-Based Enterprise. There is a common misunderstanding that digital technologies such as Product Lifecycle Management (PLM) software and Systems Modeling Language (SysML) software environments can be purchased to automatically create the Digital Thread or the Model-Based Enterprise.

Digital technologies can serve as mechanisms for connecting the Digital Thread and creating the Model-Based Enterprise, but they themselves do not provide methods for analytical methods to systematically capturing hidden processes and data and information flows down to the data element level view. Therefore, the second step of the methodology for this research is to address Research Question 2: could a method for the standardized capture, mapping, and analysis of data flows composed of data elements be utilized to realize the Digital Thread and the Model-Based Enterprise? Research Question 2 will be answered by applying the analytical tool developed to address Research Question 1 across multiple domains.

The first application is to an existing product realization process of a prototyping organization focused on enabling the Digital Thread. The second application is the development of a new process for an organization that is developing verification and validation processes for modeling and simulation of systems. This second application is focused on enabling the Model-Based Enterprise through Model-Based Systems Engineering. This methodology ensures that the tool developed to address Research Question 1 is generalizable, repeatable, suitable for analyzing new and existing systems, and able to account for differences and hidden processes in real-life systems.

4.3 Addressing Research Question 3

The third step of the methodology for this research is to address research question 3: can the efficiency of the data flows captured using such a method be quantitatively measured and systematically improved? Research Question 3 will be accomplished by utilizing the results captured from the first application of the analytical tool to the product realization process in the prototyping organization. A systematic continuous improvement approach is developed to take the initial results from the first application of the analytical tool, identify sub-optimal data flows, and re-organize them into an improved state. This methodology is necessary to validate the DEMA methodology and to contribute to the business case for digitalization.

Past work in data and information waste quantification by Yarbrough [112] and Roh et al. [76] is evaluated for incorporation into the approach. New metrics for data and information flows will be developed as needed. A sample of an improved data architecture will be developed using the continuous improvement approach. Research has shown that fully enacting the digital thread is both costly and time consuming [113]. Therefore, the method presented for creating an improved digital system architecture has to prioritize identifying and improving the system's critical data threads based on data element reuse and potential metrics of improvement.

Chapter 5

Development of Data Element Mapping and Analysis (DEMA) Approach

The terms unique to DEMA are functional area, sub-functional area, functional activity, data vessel, and data element. Table 5.1 provides definitions and examples for each of the DEMA terms. The Digital Thread is defined as the connection of data and information flows throughout a system lifecycle [20][19][33][21], and this research asserts that it is the connection of data elements, and not data vessels or functions, that results in the Digital Thread. Although the term "Digital Thread" is often used in the singular form, in reality many digital threads will make up the "Digital Thread" proper as not every data element in the lifecycle is relationally connected to one another. It is also worth noting that the flow of data and information does not imply connectivity.

Table 5.1 DEMA Terminology

DEMA Term	**Definition**	**Examples**
Functional Area	Functions that represent the highest level in which actions take place within a system.	Engineering, Verification, Validation, Project Management, and Manufacturing
Sub-Functional Area	Sub-levels within functional areas.	Within an Engineering functional area, sub-functional areas could include Initial Design and Technical Data Package Assembly.
Functional Activity	The operations within functional areas that transform data vessel inputs into outputs	Within an Engineering sub-functional area Initial Design, functional activities could include Requirements Elicitation, Conceptual Design, and Conceptual Design Review.
Data Vessel	Any container used to transport data elements	Emails, personal notes, MBSE files, word documents, and software files
Data Element	The individual units of data contained within data vessels	Document titles, dimensions, software file inputs, individual requirements, and due dates

DEMA is based on system decomposition methods and elicitation techniques. Mapping techniques including Process Flow Charts, Integration Definition Methods, Control

Flow Diagrams (CFD), Data Flow Diagrams (DFD), Functional Flow Diagrams (FFD), Functional Flow Block Diagrams (FFBD), Business Process Modeling Notation (BPMN), Value Stream Mapping (VSM), Unified Modeling Language (UML), Systems Modeling Language (SysML), N^2 diagrams, and E2E Data Flow Diagrams were evaluated for suitability in isolating threads of data elements (the individual units of data such as part dimensions, meeting times, and requirements) as they move across actors and places of storage.

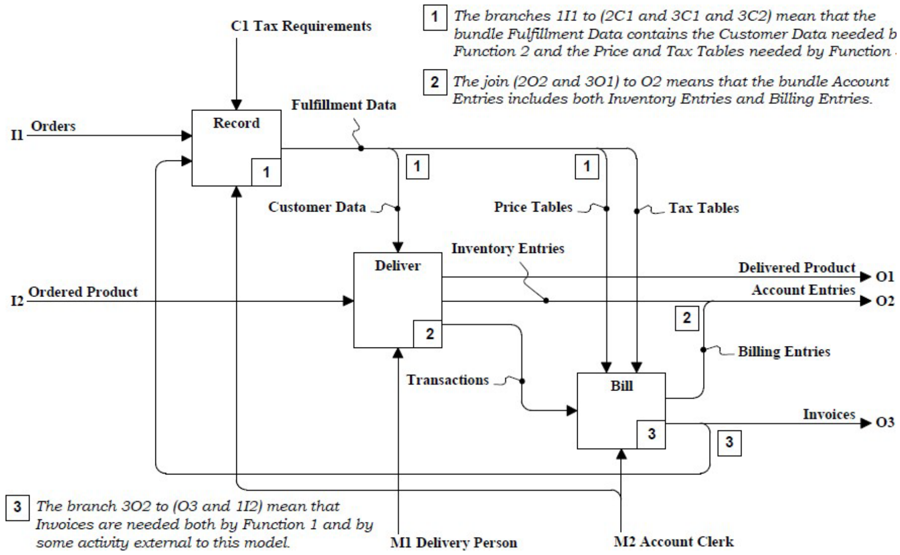


Figure 5.1: IDEF0 Example from IEEE [118]

The results of this analysis are shown in Table 5.2. The first column lists the name of the mapping technique. In columns 2-5, an “X” is placed if the mapping meets the criteria outlined in the column header. In the second column the criterion is does the technique allow for the mapping places of data storage. In the third column the criterion is does the technique allow for the mapping of actors who access data. In column four, the criterion is does the mapping allow data element thread identification to be practical (easy to read without mapping objects such as function blocks impeding the isolation of data element threads). And, in column five, the criterion is does the technique have a systematic method for uncovering hidden processes, data, and information flows that are found in real-life systems.

Table 5.2: Evaluation of Past Visual Mapping Techniques for Data Element Thread Isolation

Mapping	**IDs Storage**	**IDs Actors**	**Data Element Thread Isolation Easy to See**	**Hidden Data**
Process Flow Chart	-	-	-	-
CFD	-	-	-	-
DFD	X	X	-	-
FFD	-	-	-	-
FFBD	-	-	-	-
BPMN	X	X	-	-
VSM	X	X	-	-
N2	-	-	-	-
IDEF0	-	-	-	-
IDEF1	X	X	-	-
IDEF1X	X	X	-	-
IDEF3	X	X	-	-
IDEF4	X	X	-	-
IDEF5	-	-	-	-
SysML	X	X	-	-
UML	X	X	-	-
E2E	X	X	-	X

Knowledge Based Systems, Inc maintains the sixteen types of IDEF, [114], but only six of their IDEF models are defined and made available to the public on their website [115]. Therefore, only these six were included in the analysis. Within SysML, there are many different types of diagrams [116], and the same is true for UML [117]. Therefore, if any of the mappings available within UML and SysML met the capability criteria laid out in the columns of Table 5.2, then they were marked as having met those capabilities.

Two major issues were quickly discovered when analyzing each of the mapping techniques. The first was that although several of the mapping techniques could be used to visualize the flow of documents, their functional and hierarchical perspective inhibited their ability to isolate threads of data. For reference, see the example IDEF0 mapping in Figure 5.1 by IEEE [118]. The inputs and outputs of the IDEF0 are not limited to data but can also include physical items [118]. Thus, it can become difficult when viewing an IDEF0 to determine what inputs and outputs are data and what are physical.

More importantly, the inputs and outputs on the IDEF0 that represent data do not show the data element level. In theory, an IDEF0 could be created to show only data elements as inputs and outputs but given that fact that some documents contain tens or hundreds of individual elements, this becomes impractical as the diagram would quickly become difficult to read. Also, even if the data elements were shown as inputs and outputs, there is no construct in IDEF0 that could show who accesses each data element, from what place of storage, and under what conditions. Therefore, it would be impossible to use the IDEF0 to isolate data elements to identify breaks (manual transfer of data elements) in their digital threads. This would be the same with most of the other functional mapping techniques because they all utilize functional blocks and only around half have constructs to show who accesses each data element and from which places of storage.

The functional mapping techniques that do have some constructs that could be used to show a document's place of storage and the actors accessing them are Data Flow Diagrams (DFD), Business Process Modeling Notation (BPMN), Value Stream Mapping (VSM), IDEF1, IDEF1X, IDEF3, IDEF4, SysML, E2E Data Flow Diagrams, and UML. Also, some versions of Process Flow Charts have been created to show data and information, but the five most common process symbols announced by ASME do not include data places of storage or actor access, so it is not included in this list [25]. However, each of these mapping techniques focuses on documents instead of individual units of data (i.e., data elements), and if someone tried to record data elements instead of documents, the mappings would become difficult to read. Also, it would be challenging to isolate the individual threads as the functional blocks and other diagram objects would inhibit their isolation.

The second major issue uncovered was that traditional visual mapping techniques were found to be insufficient on their own, as they do not offer a systematic method for uncovering the complexities in hidden data and information flows that are found in real-life systems. Work procedures can provide valuable information about the stakeholders, functions, major documents, and governing standards of organization activities, but they do not capture all data and information activities that drive organizational processes. Therefore, if mapping creators evaluate work procedures and conduct interviews with stakeholders that don't systematically uncover and address hidden processes in the system, it is unlikely that all hidden data and information flows will be revealed. None of the mapping techniques except for E2E Data Flow Diagrams explicitly outlined methodologies to systematically uncover hidden processes and hidden data.

Therefore, traditional mapping techniques by themselves are helpful in enabling digitization of standardized documents, but not true digitalization, which is the fundamental reorganization of existing processes to improve connectivity and information flows [7]. There are, however, some very important capabilities offered by past mapping techniques. Decomposing complex systems into smaller groups is very useful in managing complexity and promoting system understanding [15], and functional visual mapping techniques are useful in the high-level analysis of a system. The results shown in Table 5.2 reveal that past mapping techniques are suitable for evaluating functional, document-based, hierarchical, and software-centric views of data and information. These methods do not consider the individual units of data, or data elements, that are contained within documents. The functional perspectives make it difficult or impossible to identify and isolate individual threads of data across the lifecycle. The methods also do not provide methods for systematically discovering and uncovering hidden processes and data and information flows.

The lack of a tool capable of identifying all the data elements at the granularity necessary to potentially connect digital threads leads to the creation of a novel, generalizable, and repeatable tool - Data Element Mapping and Analysis (DEMA). DEMA integrates various traits of traditional functional mapping techniques with systems engineering elicitation methodologies to provide a partial view of the current state of data and information flows. DEMA enables the movement from a partial view to a holistic data element-level view of a system. The data element level view is where the data threads can be isolated to determine what needs to happen for the right people to have the right data, in the right place, at the right time, and in the right form to make the best decisions.

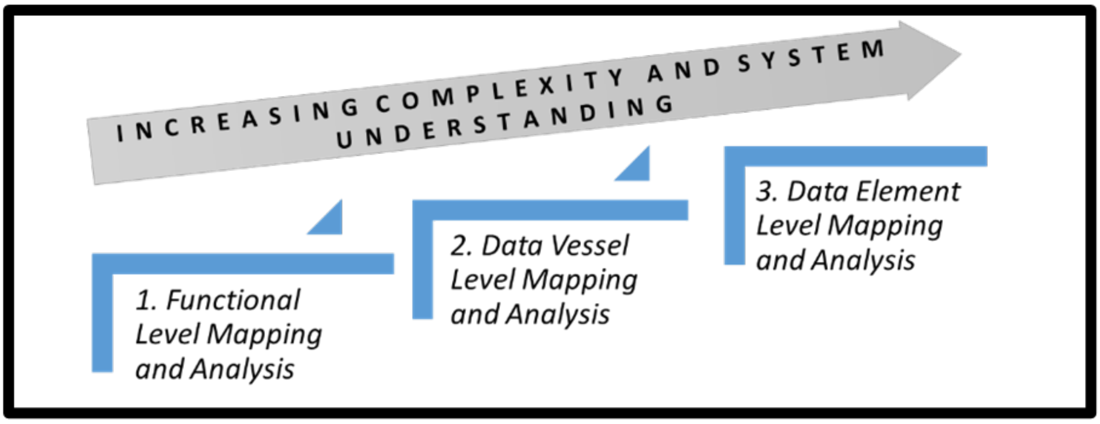


Figure 5.2: DEMA Overview

DEMA is a three-step process that enables the standardized capture, mapping, and analysis of data flows for digital system understanding and architecture development. Each step of DEMA is essential and must be carried out sequentially to achieve the end-goal Data Element Level View. The first step of DEMA is the creation of the Functional Level View, where system decomposition takes place. The second step is the Data Vessel Level View where all containers of data are identified and visually mapped. Valuable insights and opportunities for system improvement are uncovered at each step of DEMA, but the final step, the Data Element Level View, is unique in that it isolates the flow of individual data elements across the system that serve as the connectors of the Digital Thread. The steps of DEMA are outlined in Figure 5.2

As almost every system consists of a hierarchy of elements [17], DEMA can be applied to any system that involves the flow of data, with data being defined as any fact or information that can be used to make decisions, analyses, or calculations [121]. There may be concern about using DEMA if the system to be analyzed is not well understood and is without clearly defined processes or if the system in question varies greatly depending on its application. However, DEMA is very suitable for application in not well-understood systems, as DEMA offers a standardized method to uncover the complexities of such a system, and mechanisms can easily be added to each step to account for system variability.

5.1 DEMA Step 1

The first step of DEMA is Functional Level Mapping and Analysis. In this step, the high-level functional areas, sub-functional areas, and functional activities are identified and visually mapped. This is a critical DEMA process as all captured data and information flows must be traceable to high level functions, so that the data and information flows identified in DEMA Steps 1 and 2 can be traced back to organizational requirements. In this step, hidden functional activities (activities unknown to the organization) are also identified and recorded. The functions identified in Step 1 serve as a foundation to establish the flow of data vessels in Step 2, and without them, the end goal of the Data Element Level View in Step 3 cannot be achieved.

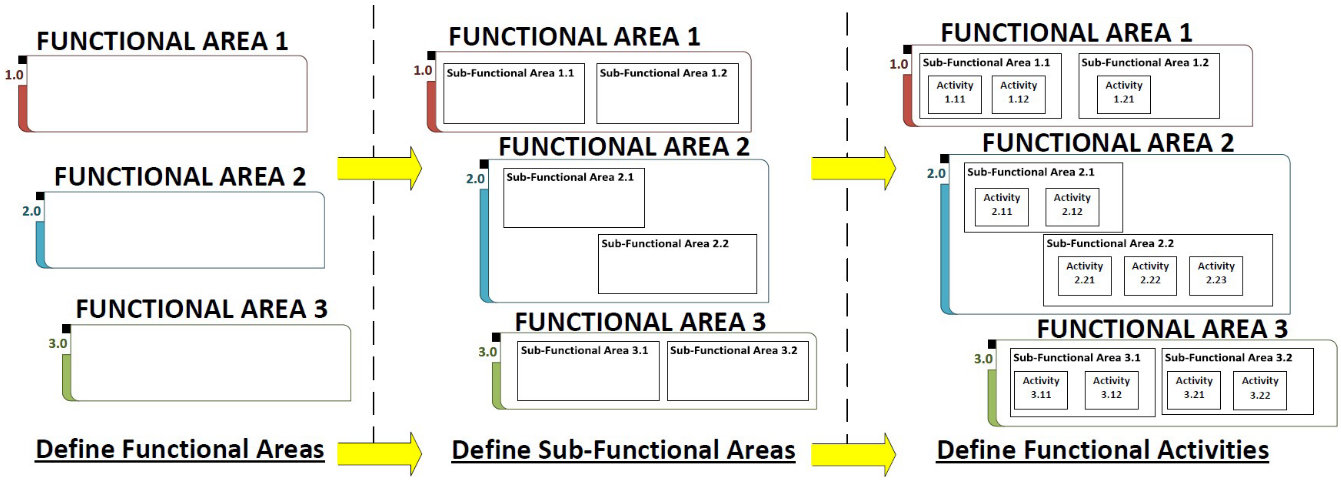


Figure 5.3: Steps to Create Functional Level Mapping

The high-level functional areas, the sub-functional areas, and functional activities can be identified using several resources. If program management data and work procedures are available, they can be used to identify the functions. Elicitation techniques and round table discussions with system stakeholders are also essential in ensuring that the captured functions accurately reflect the real-world system. Once all functions are identified, they can be visually mapped using Functional Block Diagram (FBD) and Functional Flow Block Diagrams (FFBD). A generic example of a Functional Level Mapping is shown in Figure 5.3.; the high-level functions of the system are shown on the left, the identification of the sub-functional areas is shown in the middle, and then the functional activities are filled in on the right. Once the visual mappings have been verified by the system's stakeholders, the Functional Level View and Mapping is complete.

5.2 DEMA Step 2

The second step in DEMA is the Data Vessel Level Mapping and Analysis. In this step, a visual mapping is created that shows the flow of data vessels across the functional areas, sub-functional areas, and functional activities identified in Step 1. Data vessels are documents, emails, conversations, personal notes, drawings, CAD files, and any other possible container (i.e., vessel) of data. Although the Functional Level View in Step 1 is essential to gain insight into the system and to establish a foundation for the flow of data vessels, it does not address the data and information exchanges between the functional activities. Therefore, Step 2 of DEMA must be conducted to identify the flow of data vessels so that full digitalization is enabled.

At the beginning of Step 2, representatives from each of the Functional Areas are presented with the initial Functional Level Mapping from Step 1. They are asked to identify which of the high-level functional areas and functional activities that they are involved in and to provide descriptions for the activities in the Functional Mapping. For systems where the functional activities are not clearly defined and standardized within the system, the interviewee should first be asked to describe the activities in which they participate in the system and their responses recorded.

It may be helpful for the interviewer to start by asking about the activities at the end of the process and then work backwards from there to have the interviewee define the process. This method of working backwards through a process is deployed when conducting value stream mapping [119]. The idea of using standardized elicitation techniques to develop visual mappings has been proven successful in various instances. Systemigrams have been created by deploying a ten-step process to capture, characterize, and visually map the views of stakeholders for enterprise analysis [22]. Research discussed in Section 2.2 mapped information flows using value stream mapping techniques that employed interviews for data capture [76] [77].

Once the interviewee identifies the functions in which they were involved, they are asked to report the data vessels that are the inputs and outputs of each functional activity. Information on each data vessel's place of storage, form, means of transfer, and the actors involved in handling the data are recorded in a standardized questionnaire template. The standardized questionnaire template is provided in the appendix.

Important knowledge that can be reused throughout the product lifecycle is often not captured in an easily searchable digital form [97], and the ultimate goal for data and information flows is to enable the right people to have the right data, in the right place, at the right time, and in the right form to make the best decision [14]. Therefore, it is important that this information is recorded in the template so that opportunities for improvement can be identified. From there, the interview results are visually mapped using a technique derived from an integration of IDEF0 and DeMarco's Data Flow Diagrams to record the flow, storage, and access of data vessels across the functions. Figure 5.4 shows the syntax used to create a data vessel mapping, and Figure 5.5 shows an example of a generic data vessel mapping for a single activity.

In a Data Vessel Map, the boxes represent the functional activities, and the name of each data vessel is recorded on the arrows going into and out of the functional activity boxes. When a specific data vessel flows either within or across functions as an input to become an output, or vice versa, this is considered a data vessel dependency. The labels on arrows going into the boxes are data vessel inputs to the function, and those going out are the data vessel outputs. The actors that handled the data are recorded on the arrows going into the top and bottom of the functional activity boxes (there is no difference in actors based on whether the arrow is at the top or bottom of the function box).

The data vessel retrieval or storage locations are indicated by the cylindrical database shape The data vessels are connected to their place of retrieval/storage by lines connected to brackets grouping data vessels to their respective place of retrieval/storage. When a data vessel is connected to an "X" it means that the data vessel was not stored in an official place of storage. When a data vessel is connected to a highlighted "???", it means that the place of storage or retrieval was unknown. For this application, mappings were created in Microsoft Visio for each of the six functional areas.

One finding from the interviews may be that different actors within the system conduct different patterns of data exchanges while performing the same functions within an activity. In these instances, separate mappings should be created for each actors' process. Differences such as this may reveal a need for standardization in activities and data exchanges. However, DEMA does not stop at the Data Vessel Level View and Mapping. It is imperative that a document- centric and approval-based view of a data and information flow must be overcome to enable digitalization [79][120] since data elements are contained within the data vessels, and it is the data elements, not the data vessels, that matter. The data vessels are instrumentally important to the extent that they enable organization, traceability, security, and appropriate access to the data elements. Therefore, a Data Element Level View is necessary to enable digitalization.

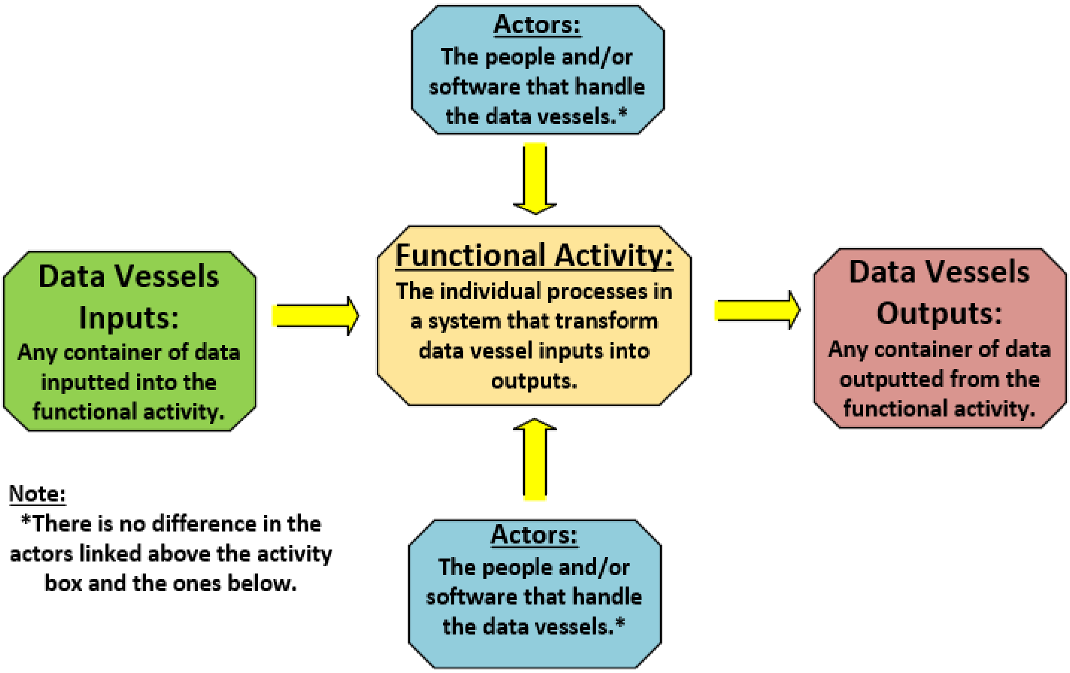


Figure 5.4: Data Vessel Mapping Syntax

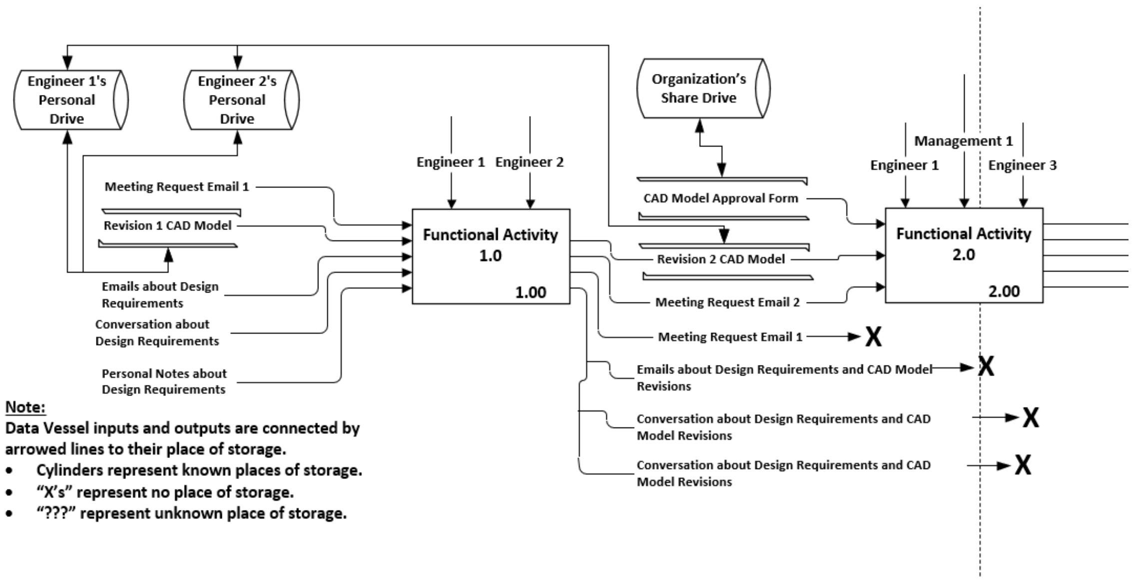


Figure 5.5: Generic Data Vessel Mapping For Single Activity

5.3 DEMA Step 3

The Data Element Level View is the final step of DEMA. The output of this step is the identification, capturing, and listing the Data Elements within each data vessel as they flow throughout the system functions. Although the Data Vessel Level View captured in Step 2 provides insight into the containers (i.e., vessels) of data, they do not even begin to address the data elements that would make up the connectors of the data and information flows.

Also, the Data Vessel Level View only shows what data vessels are required as inputs and outputs into each function instead of all data exchanges and actor access that would occur in the real-world system. Therefore, the Data Element Level View is necessary to determine the system requirements and architecture to build the Digital Thread or other digital system such as the Model-Based Enterprise.

The Data Vessel Level Mapping in Step 2 recorded each of the data vessels involved in the system and where they came from and were stored. Therefore, the Data Vessel Level Mapping can be used to seek and retrieve the data elements contained within each of the data vessels. From there, the flow of data elements can be listed and visually mapped using the flows identified in the Data Vessel Level Mapping. The Data Element Level View captures the current state of data and information flows to the level of detail necessary to determine what must happen for the right people to access the right information, at the right time, and in the right form.

Data elements themselves are the smallest units of data contained within a vessel. For example, an engineering drawing may include data elements including but not limited to dimensions, tolerances, revision numbers, approval signatures, date of approvals, and drawing titles. An emailed meeting invitation could include data elements such as email addresses, name of sender, name of receiver, meeting date, meeting time, meeting location, and other texts. In Figure 5.6, an example is shown of data elements being identified in a generic form used by an organization for fabrication.

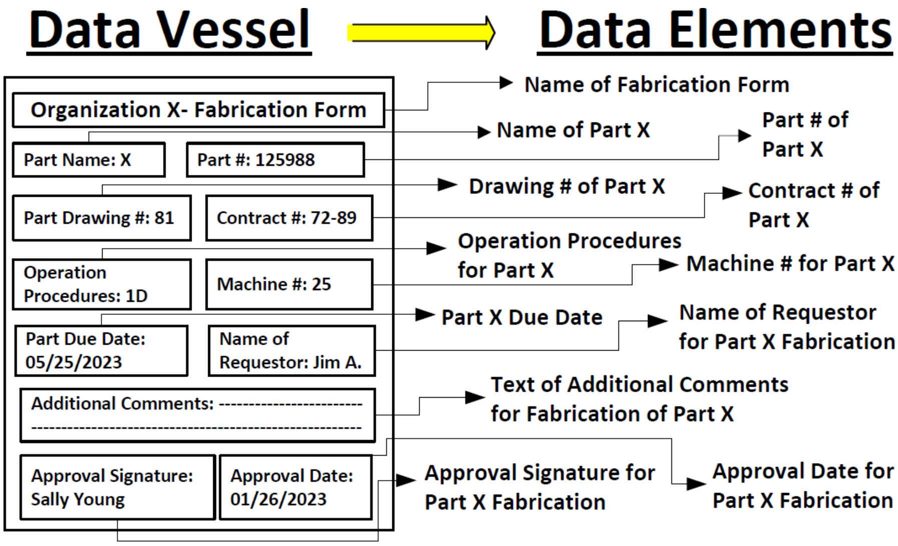


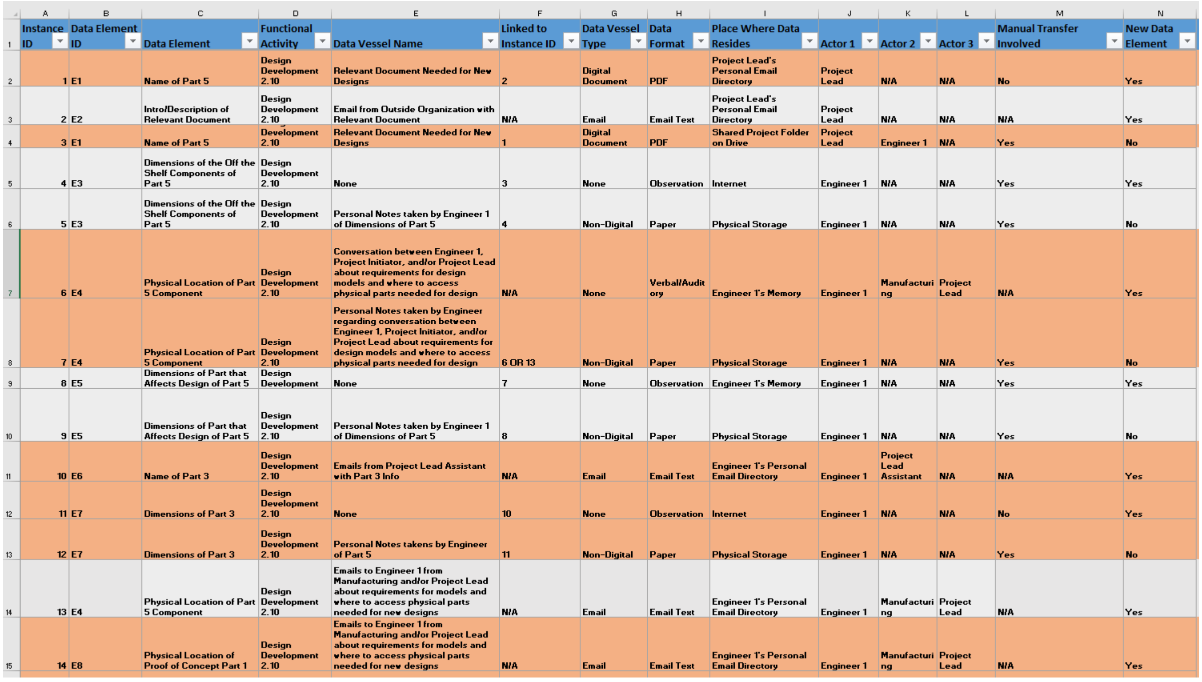
Figure 5.6: Generic Example of Data Element Identification

The flow of data elements are recorded in Microsoft Excel. Each row in Excel corresponds to one instance of data element access by an actor in the lifecycle. The Instance ID in Column A and the Data Element ID in Column C are assigned manually. The data shown in each column in the Excel Sheet are shown in Table 5.3. There may be cases where it is most practical to handle data elements aggregately in the Data Element Excel Sheet. For example, if a data vessel is used that is mostly text-based such as a report, an entire chapter of the report including all of its sub-section could be treated as an aggregate data element.

The purpose of the Data Element Level View Listing in Excel is to isolate and identify individual threads of data. Figure 5.7 shows several rows of the Data Element Excel Sheet, and Figure 5.8 shows an example of isolating the flow of an individual data element by using the filter function on the Data Element Column C. This allows visualization of how the data element is passed throughout the lifecycle between actors, functions, and data vessels, and the visualization of how much of the data flow is driven by manual intervention such as email exchanges, moving data elements from paper notes to CAD models, and retrieving data from various places of storage. From another perspective, it shows where data is currently stored and retrieved as well as which data elements are not stored at all.

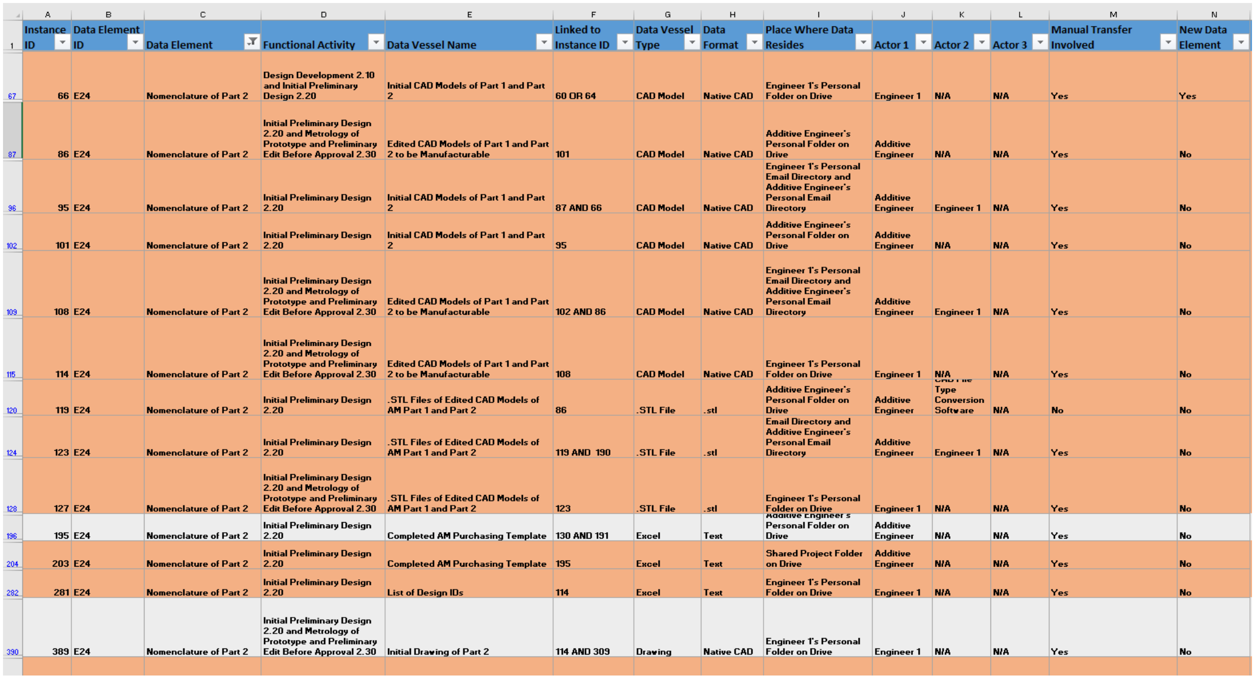
Table 5.3 Definition of Column Data in Data Element Level View Excel Sheet

Column Name	**Definition of Data**
Instance ID	A unique identifier for the data element instance.
Data Element ID	A unique identifier for the data element accessed in the instance.
Data Element	The name of the data element accessed in the instance.
Functional Activity	The functional activity under which the instance takes place.
Data Vessel Name	The name of the data vessel (as listed in the Data Vessel Mapping) in which the data element is accessed in that instance.
Link to Instance ID	The Instance ID of data element access instance that directly precedes the current data element instance.
Data Vessel Type	The type of data vessel in which the data element is accessed in that instance (Email, Digital Document, Non-Digital, CAD Model, etc.).
Data Format	The format of the Data Vessel in which the data element is accessed in that instance (Text, PDF, Excel, Paper, Native CAD, etc.)
Place Where Data Resides	The place of storage where the Data Vessel in which the data element resides and is accessed in that instance.
Actors 1, 2, and 3	The first, second, and third person, software, or any other possible actor who is accessing and/or handling the data element in that instance.
Manual Transfer Involved?	A "Yes" or "No" answer as to whether the transfer between the instance of data element access that directly preceded the current data element instance (referenced in the "Link to Instance ID" column) was manual (transferred between human actors). A "No" answer indicates that some form of digital connectivity exists between the instances.
New Data Element	A "Yes" or "No" answer as to whether the transition between the instance of data element access that directly preceded the data element instance (referenced in the "Link to Instance ID" column) introduces a new data element.



	Instance ID	Data Element ID	Data Element	Functional Activity	Data Vessel Name	Linked to Instance ID	Data Vessel Type	Data Format	Place Where Data Resides	Actor 1	Actor 2	Actor 3	Manual Transfer Involved	New Data Element
2	1	E1	Name of Part 5	Design Development 2.10	Relevant Document Needed for New Designs	2	Digital Document	PDF	Project Lead's Personal Email Directory	Project Lead	N/A	N/A	No	Yes
3	2	E2	Intro/Description of Relevant Document	Design Development 2.10	Email from Outside Organization with Relevant Document	N/A	Email	Email Text	Project Lead's Personal Email Directory	Project Lead	N/A	N/A	N/A	Yes
4	3	E1	Name of Part 5	Development 2.10	Relevant Document Needed for New Designs	1	Digital Document	PDF	Shared Project Folder on Drive	Project Lead	Engineer 1	N/A	Yes	No
5	4	E3	Dimensions of the Off the Shelf Components of Part 5	Design Development 2.10	None	3	None	Observation	Internet	Engineer 1	N/A	N/A	Yes	Yes
6	5	E3	Dimensions of the Off the Shelf Components of Part 5	Design Development 2.10	Personal Notes taken by Engineer 1 of Dimensions of Part 5	4	Non-Digital	Paper	Physical Storage	Engineer 1	N/A	N/A	Yes	No
7	6	E4	Physical Location of Part 5 Component	Design Development 2.10	Conversation between Engineer 1, Project Initiator, and/or Project Lead about requirements for design models and where to access physical parts needed for design	N/A	None	Verbal/Auditory	Engineer 1's Memory	Engineer 1	Manufacturing	Project Lead	N/A	Yes
8	7	E4	Physical Location of Part 5 Component	Design Development 2.10	Personal Notes taken by Engineer regarding conversation between Engineer 1, Project Initiator, and/or Project Lead about requirements for design models and where to access physical parts needed for design	6 OR 13	Non-Digital	Paper	Physical Storage	Engineer 1	N/A	N/A	Yes	No
9	8	E5	Dimensions of Part that Affects Design of Part 5	Design Development	None	7	None	Observation	Engineer 1's Memory	Engineer 1	N/A	N/A	Yes	Yes
10	9	E5	Dimensions of Part that Affects Design of Part 5	Design Development 2.10	Personal Notes taken by Engineer 1 of Dimensions of Part 5	8	Non-Digital	Paper	Physical Storage	Engineer 1	N/A	N/A	Yes	No
11	10	E6	Name of Part 3	Design Development 2.10	Emails from Project Lead Assistant with Part 3 Info	N/A	Email	Email Text	Engineer 1's Personal Email Directory	Engineer 1	Project Lead Assistant	N/A	N/A	Yes
12	11	E7	Dimensions of Part 3	Design Development 2.10	None	10	None	Observation	Internet	Engineer 1	N/A	N/A	No	Yes
13	12	E7	Dimensions of Part 3	Design Development 2.10	Personal Notes takens by Engineer of Part 5	11	Non-Digital	Paper	Physical Storage	Engineer 1	N/A	N/A	Yes	No
14	13	E4	Physical Location of Part 5 Component	Design Development 2.10	Emails to Engineer 1 from Manufacturing and/or Project Lead about requirements for models and where to access physical parts needed for new designs	N/A	Email	Email Text	Engineer 1's Personal Email Directory	Engineer 1	Manufacturing	Project Lead	N/A	Yes
15	14	E8	Physical Location of Proof of Concept Part 1	Design Development 2.10	Emails to Engineer 1 from Manufacturing and/or Project Lead about requirements for models and where to access physical parts needed for new designs	N/A	Email	Email Text	Engineer 1's Personal Email Directory	Engineer 1	Manufacturing	Project Lead	N/A	Yes

Figure 5.7: Data Element Excel Sheet



	Instance ID	Data Element ID	Data Element	Functional Activity	Data Vessel Name	Linked to Instance ID	Data Vessel Type	Data Format	Place Where Data Resides	Actor 1	Actor 2	Actor 3	Manual Transfer Involved	New Data Element
67	66	E24	Nomenclature of Part 2	Design Development 2.10 and Initial Preliminary Design 2.20	Initial CAD Models of Part 1 and Part 2	60 OR 64	CAD Model	Native CAD	Engineer 1's Personal Folder on Drive	Engineer 1	N/A	N/A	Yes	Yes
87	86	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20 and Metrology of Prototype and Preliminary Edit Before Approval 2.30	Edited CAD Models of Part 1 and Part 2 to be Manufacturable	101	CAD Model	Native CAD	Additive Engineer's Personal Folder on Drive	Additive Engineer	N/A	N/A	Yes	No
96	95	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20	Initial CAD Models of Part 1 and Part 2	87 AND 66	CAD Model	Native CAD	Engineer 1's Personal Email Directory and Additive Engineer's Personal Email Directory	Additive Engineer	Engineer 1	N/A	Yes	No
102	101	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20	Initial CAD Models of Part 1 and Part 2	95	CAD Model	Native CAD	Additive Engineer's Personal Folder on Drive	Additive Engineer	N/A	N/A	Yes	No
109	108	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20 and Metrology of Prototype and Preliminary Edit Before Approval 2.30	Edited CAD Models of Part 1 and Part 2 to be Manufacturable	102 AND 86	CAD Model	Native CAD	Engineer 1's Personal Email Directory and Additive Engineer's Personal Email Directory	Additive Engineer	Engineer 1	N/A	Yes	No
115	114	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20 and Metrology of Prototype and Preliminary Edit Before Approval 2.30	Edited CAD Models of Part 1 and Part 2 to be Manufacturable	108	CAD Model	Native CAD	Engineer 1's Personal Folder on Drive	Engineer 1	N/A	N/A	Yes	No
120	119	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20	.STL Files of Edited CAD Models of AM Part 1 and Part 2	86	.STL File	.stl	Additive Engineer's Personal Folder on Drive	Additive Engineer	Type Conversion Software	N/A	No	No
124	123	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20	.STL Files of Edited CAD Models of AM Part 1 and Part 2	119 AND 190	.STL File	.stl	Email Directory and Additive Engineer's Personal Email Directory	Additive Engineer	Engineer 1	N/A	Yes	No
128	127	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20 and Metrology of Prototype and Preliminary Edit Before Approval 2.30	.STL Files of Edited CAD Models of AM Part 1 and Part 2	123	.STL File	.stl	Engineer 1's Personal Folder on Drive	Engineer 1	N/A	N/A	Yes	No
196	195	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20	Completed AM Purchasing Template	130 AND 191	Excel	Text	Additive Engineer's Personal Folder on Drive	Additive Engineer	N/A	N/A	Yes	No
204	203	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20	Completed AM Purchasing Template	195	Excel	Text	Shared Project Folder on Drive	Additive Engineer	N/A	N/A	Yes	No
282	281	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20	List of Design IDs	114	Excel	Text	Engineer 1's Personal Folder on Drive	Engineer 1	N/A	N/A	Yes	No
390	389	E24	Nomenclature of Part 2	Initial Preliminary Design 2.20 and Metrology of Prototype and Preliminary Edit Before Approval 2.30	Initial Drawing of Part 2	114 AND 309	Drawing	Native CAD	Engineer 1's Personal Folder on Drive	Engineer 1	N/A	N/A	Yes	No

Figure 5.8: Data Thread Isolated on Data Element Excel Sheet

5.4 Results from DEMA Development

Previously, no analytical approach existed that allows for the systematic capture of hidden processes and data and information flows while uncovering progressively complex views of the system down to the data element level view, and the development of DEMA filled this research gap. The development of DEMA also proved that the analysis of data and information flows could be achieved so that the fundamental unit of the data element is identified, documented, and visualized.

Chapter 6

Application of DEMA Across Multiple Domains

The DEMA methodology was applied across multiple domains to prove the generality of the approach and to prove its effectiveness in enabling the Digital Thread and the Model-Based Enterprise. All parts of the DEMA approach, as outlined in the previous subsection, were applied across two different settings. DEMA is suitable for analyzing all types of systems with data, regardless of the current level of system understanding, digitalization, and variability. The first application was conducted within a product realization environment at a prototyping organization with a particular focus on the realization of the Digital Thread.

The second application was in an operational environment for the verification and validation of modeling and simulation applications. Whereas the first application was performed on an existing system, the second application of the DEMA methodology was employed in the development of a new system to enable the Model-Based Enterprise. The results of both applications were analyzed, and the findings were compared. The DEMA methodology itself was refined with each application. Each application of DEMA captured the current state of data and information flows at the data element level in each organization. The threads of data elements captured were analyzed, and the disconnects in the data flows were identified.

6.1 Results from DEMA Application in Product Realization Environment

In the application presented here, DEMA was applied to a product realization setting within a prototyping organization with high-mix, low-volume product types with both engineering and production functions. Within such an environment, proper management of lead times, process milestones, quality, operational costs, and flexibility are crucial to maintaining a competitive edge [122]. It is also important to remember that the product realization process itself incorporates process design, execution, and improvement [123]. Such complexity and variability in a product realization environment provides many opportunities for the identification and elimination of non-valued added activities [124].

Knowledge management, the management and use of data and information, is a critical enabler of competitiveness in product realization environments [97][110]. The actors in an organization hold knowledge of both products and the processes used to create products [111]. In practice, organizations rely heavily on highly skilled employees utilizing siloed tribal knowledge to accomplish functional activities. Such systems contain great risks and are not sustainable as they rely on a system based on heroic effort from its employees, dependent on employees doing the right thing every time and remaining indefinitely within the organization.

DEMA was applied with the intent to identify opportunities for improvement in data and information flows, to assist in developing well-documented standardized processes, and to reduce the cost and lead time for getting finished products to the customer. In situations where a complex system such as a product realization environment is analyzed with DEMA, there will almost always be differences in how the different actors within the system conduct activities with different patterns of data exchanges within each activity. In these instances, the differences should be recorded at each step of DEMA.

The overall findings from the first application were surprising. Even with detailed work procedures provided by the prototyping organization, over 90% of the data vessel handling and exchanges were nonstandard (ungoverned) and undocumented (unknown by the organization) and driven by the tribal knowledge of the actors in the system. Approximately 88% of all data element instances involved manual handling and transfer, indicating a lack of connectivity (breaks in the digital thread). Because over 25,000 data element instances were identified in the product realization process, the magnitude of the need for digital connectivity is apparent.

In addition, just under 75% of the data vessel handling and exchanges involved unstructured data vessels - that is, data vessels not in a format amenable to having their data elements digitally connected. This application showed that the results from each step of DEMA captured the current state of data and information flows in a way that allowed nonstandard (hidden) and unconnected data vessels and elements to be identified. Due to the nature of the research, and the request for anonymity from the organization, a full set of supporting data is not available. Example mappings with sensitive data removed are shown in Figures 6.1 – 6.22, and they fully demonstrate the methodology and results.

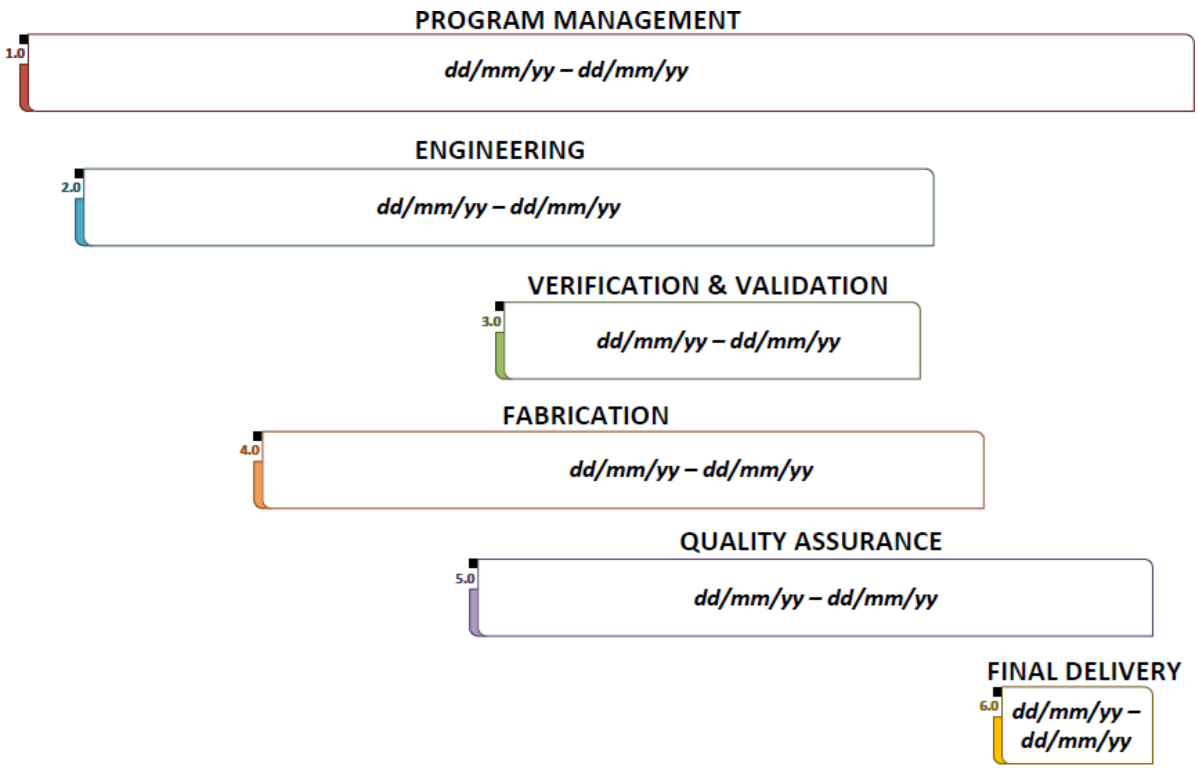


Figure 6.1: High-Level Functional Areas of the Additive Manufacturing Project

6.1.1 Step 1: Functional Level View and Analysis

The sub-functional areas and functional activities within each functional area were then determined from the IMS and mapped using Functional Block Diagrams (FBD) (see Figures 6.2 and 6.3). The IMS was also used to determine the process flow between the functional activities and mapped using Functional Flow Block Diagrams (FFBD) (see Figure 6.4). The initial Functional Level View and Analysis revealed a total of 65 functional activities, with Fabrication having the most activities at 27. Moving from six functional areas to 65 functional activities uncovered increasing levels of system complexity. After including functional flow, 77 dependencies were identified between the activities, with Fabrication having the most dependencies at 43.

The Functional Level View Mapping was presented to stakeholders from each of the functional areas. The stakeholders were asked to identify the functional activities they were involved in, and their answers were documented. The stakeholders were also asked if any functional activities were incorrect or missing from the mappings. These interviews found that two activities were missing, and two were placed in the wrong functional area, resulting in 67 activities. The resulting mapping is shown in Figure 6.5. The yellow boxes represent the two missing and 2 misplaced functional activities, and the red lines represent the new functional dependencies.

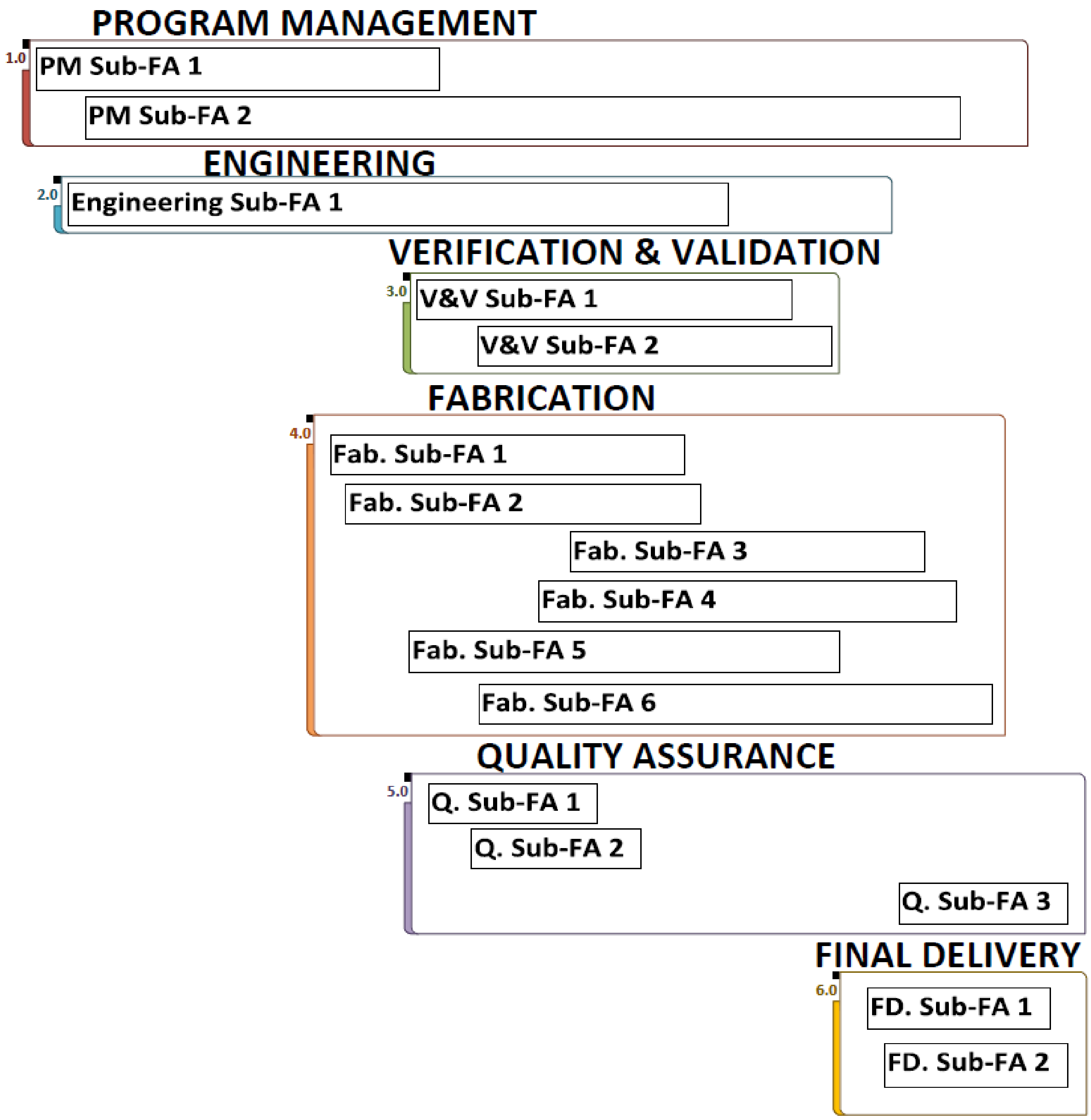


Figure 6.2: Sub-Functional Areas of the Additive Manufacturing Project

Whereas a high-level functional view of the system showed six Functional Areas, a sub-functional level view shows increasing complexity with 67 functional activities. Program management and work procedure data is helpful but not sufficient for identifying all activities, nor was it always correct. After the stakeholder interviews, four functions and three functional dependencies were corrected in the Functional Level View.

Although the Functional Level View provided insight into system processes, it did not address the data and information exchanges between the functional activities. Therefore, the results of the Functional Level View and Analysis indicated that methods are needed beyond functional analysis to capture the current state of data and information flows within a system.

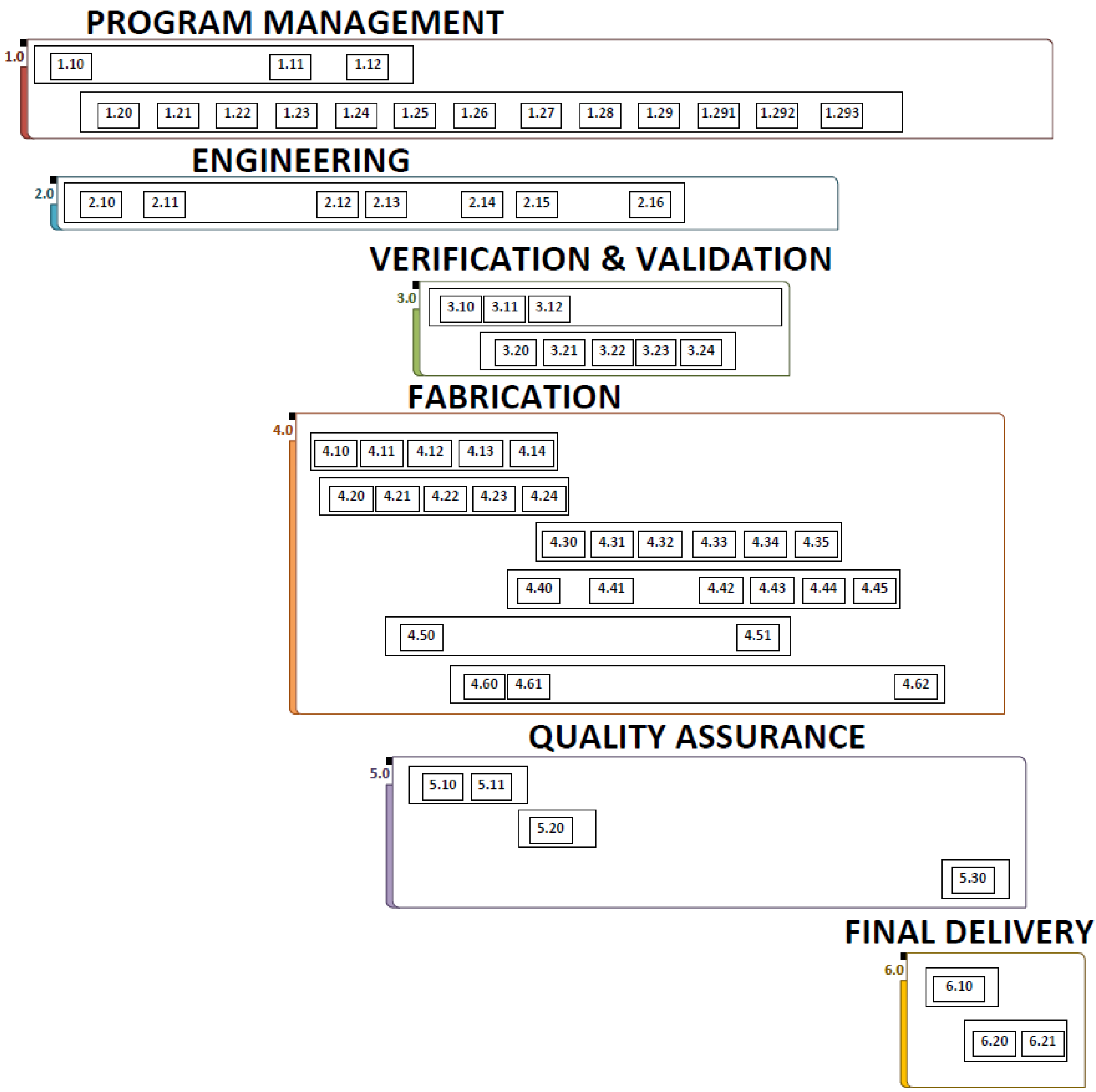


Figure 6.3: FBD of the Functional Activities of the Additive Manufacturing Project

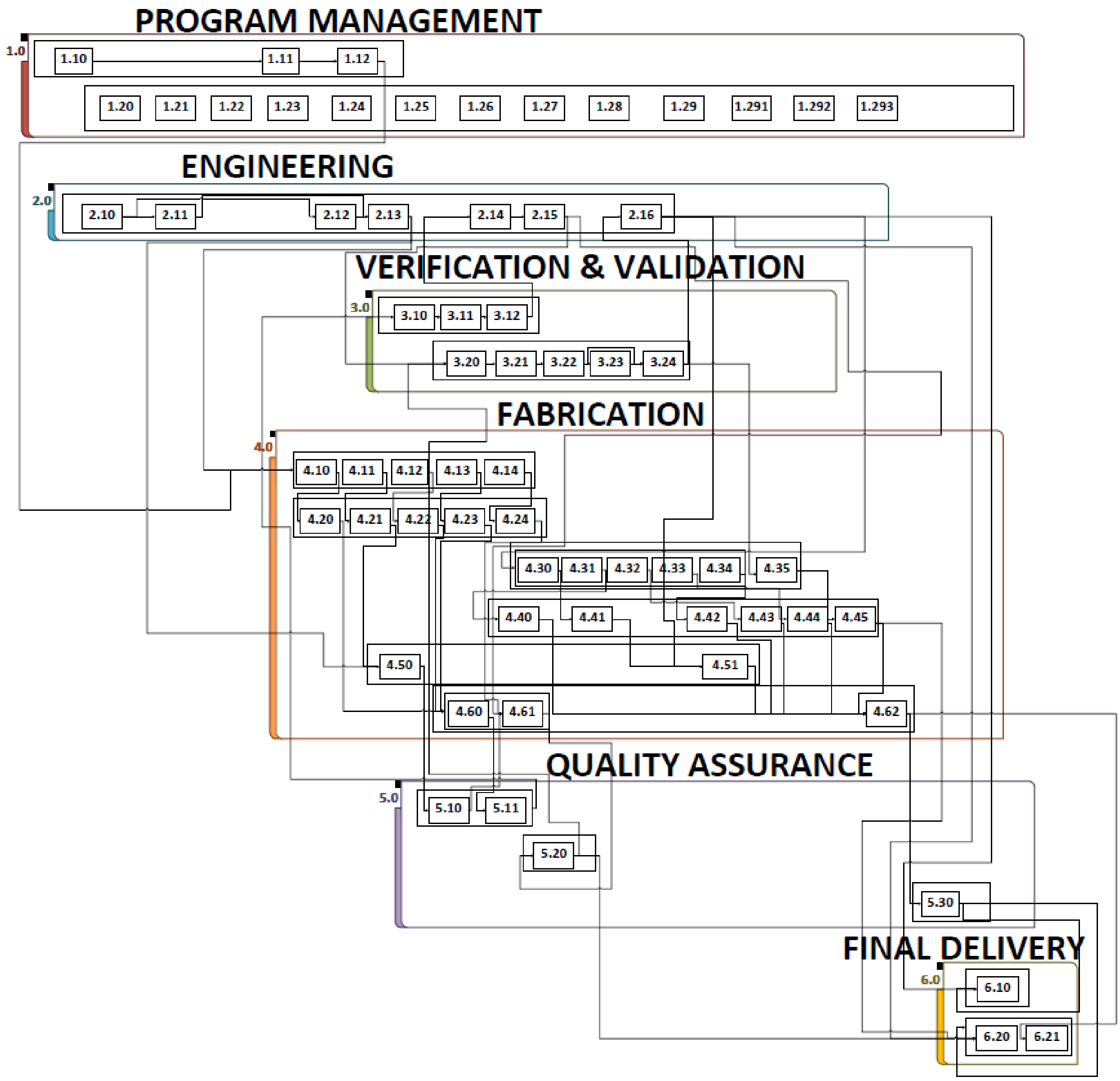


Figure 6.4: FFBD of the Functional Activities of the Additive Manufacturing Project

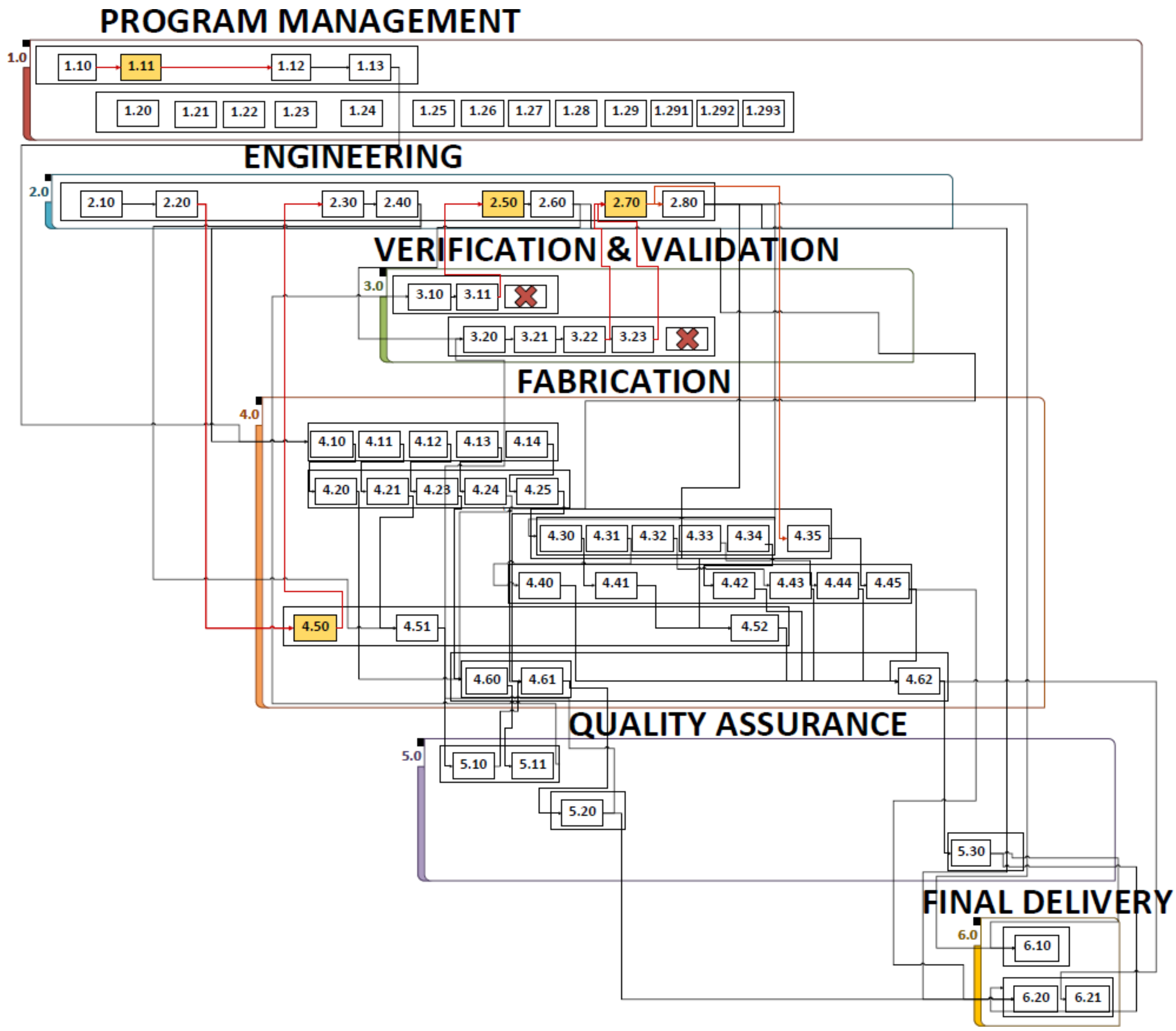


Figure 6.5: Updated FFBD of the Functional Activities of the Additive Manufacturing Project

1. Function Name and Number:

- Brief Description of Function:
  - i. Description….
- Timeline of Function: Date x/xx/xx- x/xx/xx
- Relationship to other Functions:
  - i. Preceding: Function Name/s
  - ii. Succeeding: Function Name/s
- Personnel Involved in Function and their corresponding needs in accessing data vessels:
  - i. Names/descriptions of data vessel access.
- *Qh. What Data Vessels are inputted into this activity?*
  - i. *Technical Data:*
    - *Data Vessel Input 1:*
      - a. *Data Vessel Name:*
      - b. Personnel who acquired data for activity:
      - c. Place data was retrieved from:
      - d. Format data was captured in:
      - ……..
  - ii. *Program Management Data:*
    - *Data Vessel Input 1:*
      - a. *Data Vessel Name:*
      - b. Personnel who acquired data for activity:
      - c. Place data was retrieved from:
      - d. Format data was captured in:
      - ………..
  - iii. *Business Data:*
    - *Data Vessel Input 1:*
      - a. *Data Vessel Name:*
      - b. Personnel who acquired data for activity:
      - c. Place data was retrieved from:
      - d. Format data was captured in:
      - …….
  - iv. *Other Data:*
    - *Data Vessel Input 1:*
      - a. *Data Vessel Name:*
      - b. Personnel who acquired data for activity:
      - c. Place data was retrieved from:
      - d. Format data was captured in:
      - …..
- *Qi. What Data Vessels are outputted from this activity?*
  - i. Technical Data:
    - *Data Vessel Output 1:*
      - a. Data Vessel Name:
      - b. Personnel who created and/or handled data:
      - c. Tools used to manipulate data:
      - d. Format data was captured in:
      - e. Place data was stored:
      - f. Personnel the data was sent to:
      - g. Method in which data sent to personnel:
      - …….
  - ii. Program Management Data:
    - *Data Vessel Output 1:*
      - a. Data Vessel Name:
      - b. Personnel who created and/or handled data:
      - c. Tools used to manipulate data:
      - d. Format data was captured in:
      - e. Place data was stored:
      - f. Personnel the data was sent to:
      - g. Method in which data sent to personnel:
      - …….
  - iii. Business Data:
    - *Data Vessel Output 1:*
      - a. Data Vessel Name:
      - b. Personnel who created and/or handled data:
      - c. Tools used to manipulate data:
      - d. Format data was captured in:
      - e. Place data was stored:
      - f. Personnel the data was sent to:
      - g. Method in which data sent to personnel:
      - …….
  - iv. Other Data:
    - *Data Vessel Output 1:*
      - a. Data Vessel Name:
      - b. Personnel who created and/or handled data:
      - c. Tools used to manipulate data:
      - d. Format data was captured in:
      - e. Place data was stored:
      - f. Personnel the data was sent to:
      - g. Method in which data sent to personnel:
      - …….

Figure 6.6: Data Vessel Questionnaire for Application 1

6.1.2 Step 2: Data Vessel Level Mapping and Analysis

Data Vessel Level View and Analysis was then conducted to identify the flow of data vessels across the functional activities. The Functional Level View was essential to gain insight into the processes and to establish a visual structure for the flow of data vessels. However, as previously shown, this does not reveal the data and information exchanges between the activities. The Data Vessel Level View is captured to identify the flow of data containers (i.e., vessels).

The initial Functional Level View Mapping was presented to representatives of each of the functional areas, and they were asked to identify the data vessels inputted and outputted at each of the functions in which they were involved. Their responses, as well as information on each data vessel's content, place of storage, form, means of transfer, and actors involved in handling the data, were recorded using a standardized questionnaire (see Figure 6.6).

The result of these interviews were standardized documents that recorded data vessels (as well as information about each data vessel) as they flowed through each of the functions. From there, the interview results were visually mapped using a technique derived from a combination of IDEF0 and DeMarco's Data Flow Diagrams to record the flow, storage, and access of data vessels across the high-level functions. To manage the size of the maps, individual mappings were created for each functional area (six total). Functions outside of the functional area considered are shown only to the extent that they have direct data vessel inputs or outputs into the functional area considered in each Data Vessel Mapping.

An example of the Data Vessel Mapping of the first Engineering Function is shown in Figure 6.7. Each of the six mappings were then created with the data vessel names on each input and the output arrows replaced with color coded shapes. A section of this color-coded map was created from the Engineering Functional Area and is shown in Figure 6.8. The ovals represent unstructured data, and the diamonds represent structured data. Structured data is in a format compatible for processing and analysis with digital tools, and unstructured data is not in a format compatible for such use [125]. The colors of the shapes represent the types of structured/unstructured data represented and the legend is shown in Figure 6.9. Each of the color-coded data vessel mappings from each of the functional areas are shown in Figures 6.10 - 6.17.

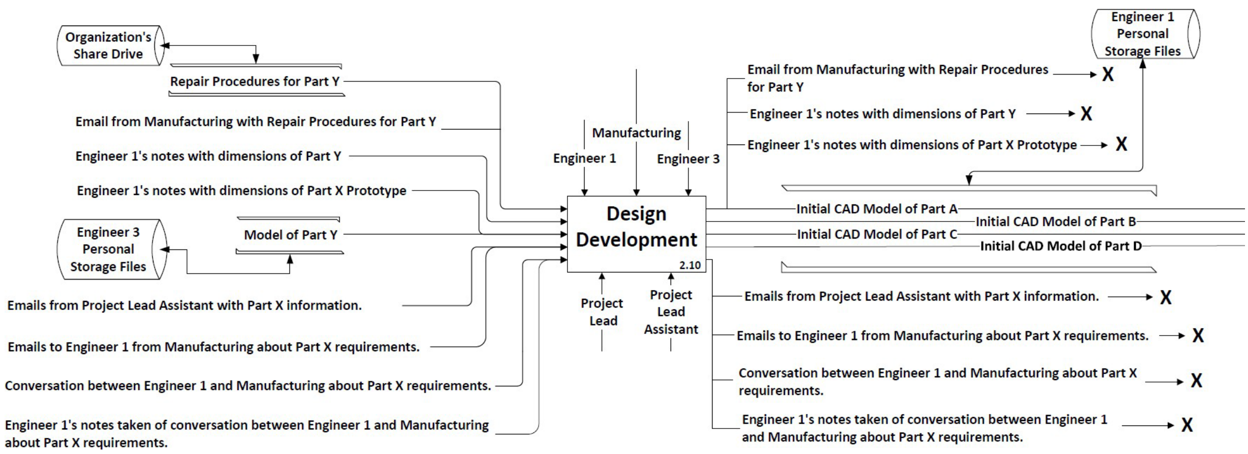


Figure 6.7: Data Vessel Mapping of First Engineering Activity

The results of the Data Vessel Level Views established a current state flow of data vessels across the organization's lifecycle functions within the functional areas. The Fabrication Function had the most data vessel inputs, outputs, and dependencies with one of the lowest percentages of structured form, standardization, and data storage for reuse. This result indicates that manufacturing systems are ideal candidates for DEMA application as they have ample opportunity for improvement.

Another interesting outcome was that although there were around 1000 data vessel inputs and outputs identified from the Data Vessel View, there were only around 500 data vessel dependencies identified. This finding implies many the data vessels did not transfer directly across the functions because they are being created and then not available throughout the lifecycle for reuse. However, even though the data vessels are not being reused, the data elements within them are being manually transferred across the lifecycle into different data vessels. Overall, the results of the Data Vessel Level View proved DEMA effective in capturing and mapping data vessel flows.

Although the Data Vessel Level View provided insight into the containers (i.e., vessels) of data, it did not address the data elements that would serve as the connectors of the Digital Thread. Also, the Data Vessel Level View only showed what data vessels were required for each function, so it did not show all data exchange and reuse that occurs in real-life for each function. Although data vessels are useful in that they can be used to organize data elements and facilitate their flow throughout a system's lifecycle, data vessels themselves are not intrinsically valuable as they do not directly create value, but they are instrumentally valuable.

Therefore, a data element view is necessary to isolate and identify the individual threads of data that actors directly use to make decisions that create value within a system. The final step, Data Element Level View, provides the identification, capture, and listing of the data elements within each data vessel as they flow through the system functions.

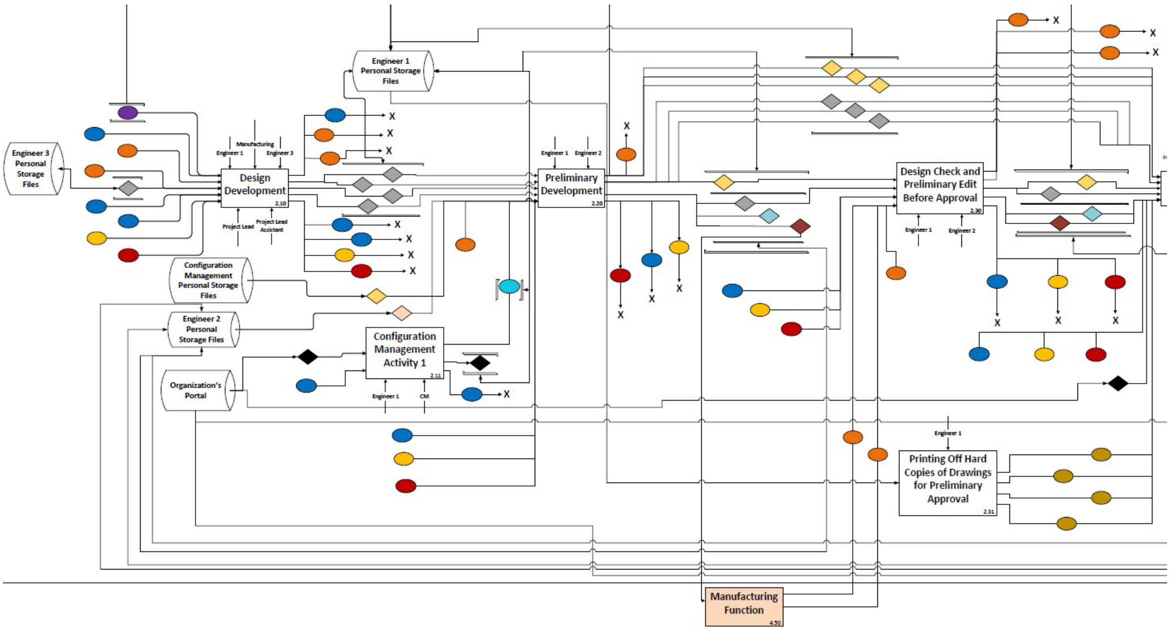


Figure 6.8: Section of Color-Coded Engineering Data Vessel Mapping

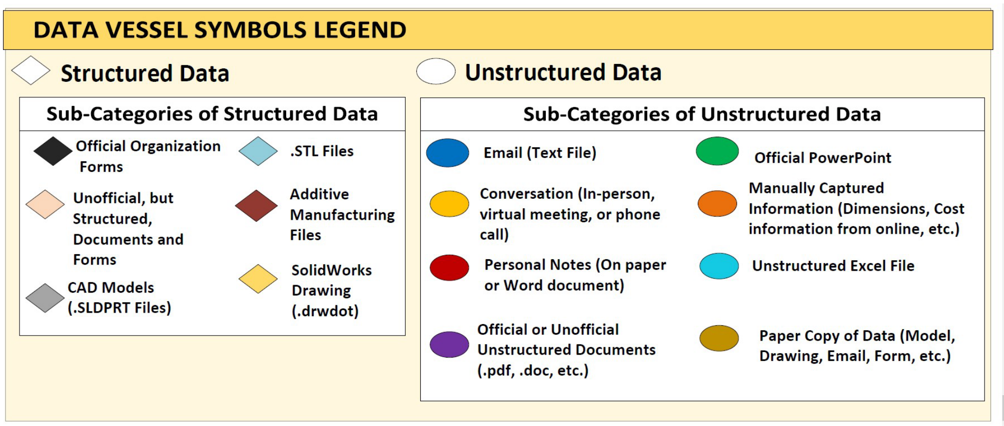


Figure 6.9: Legend for Color-Coded Data Vessel Mappings

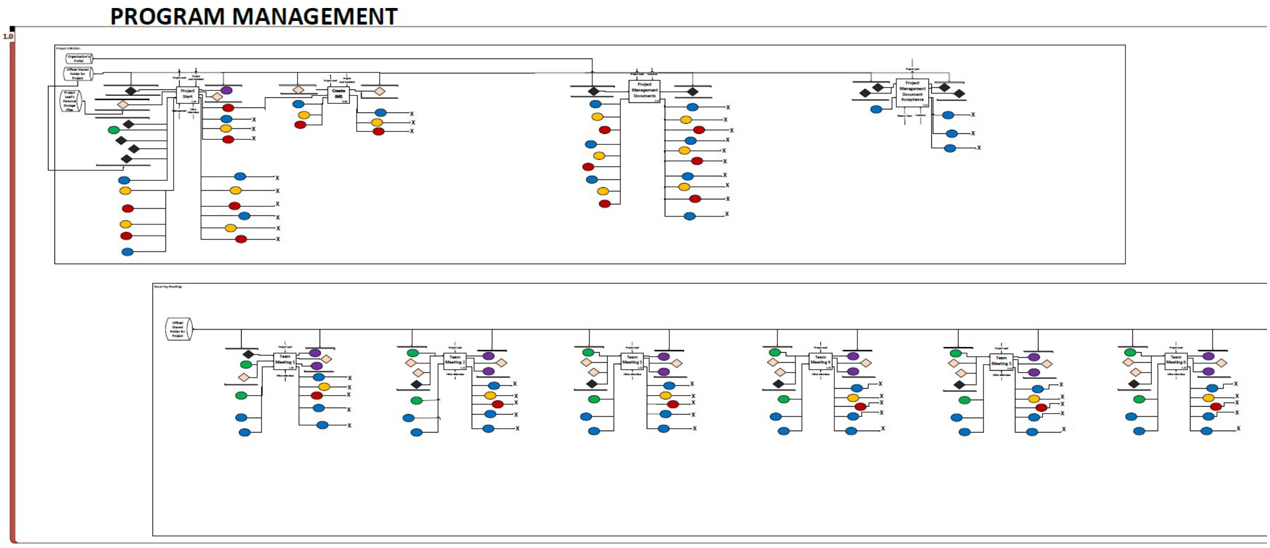


Figure 6.10: Program Management Color-Coded Data Vessel Mapping (Part 1)

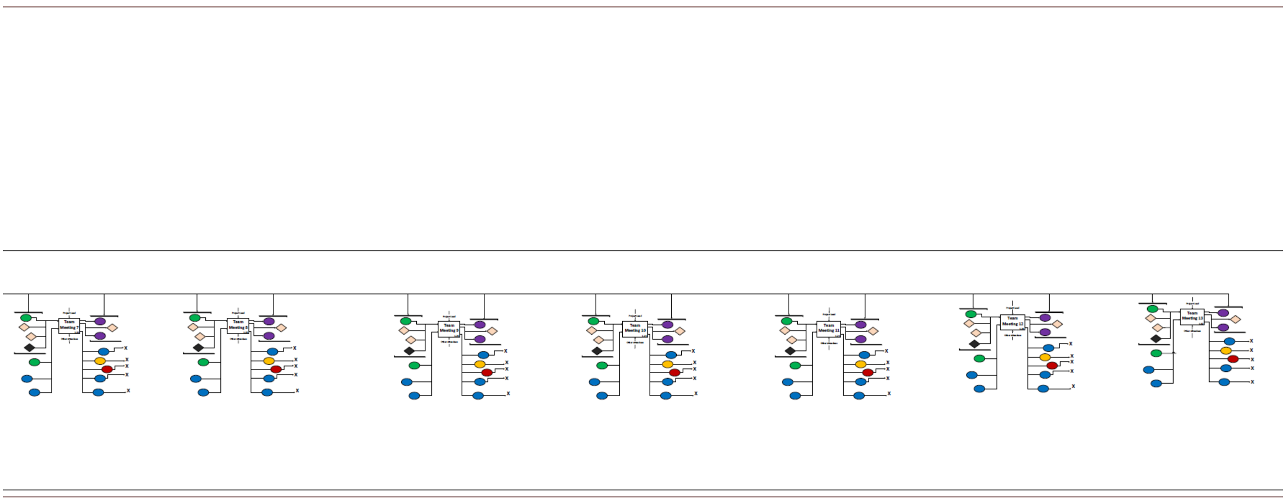


Figure 6.11: Program Management Color-Coded Data Vessel Mapping (Part 2)

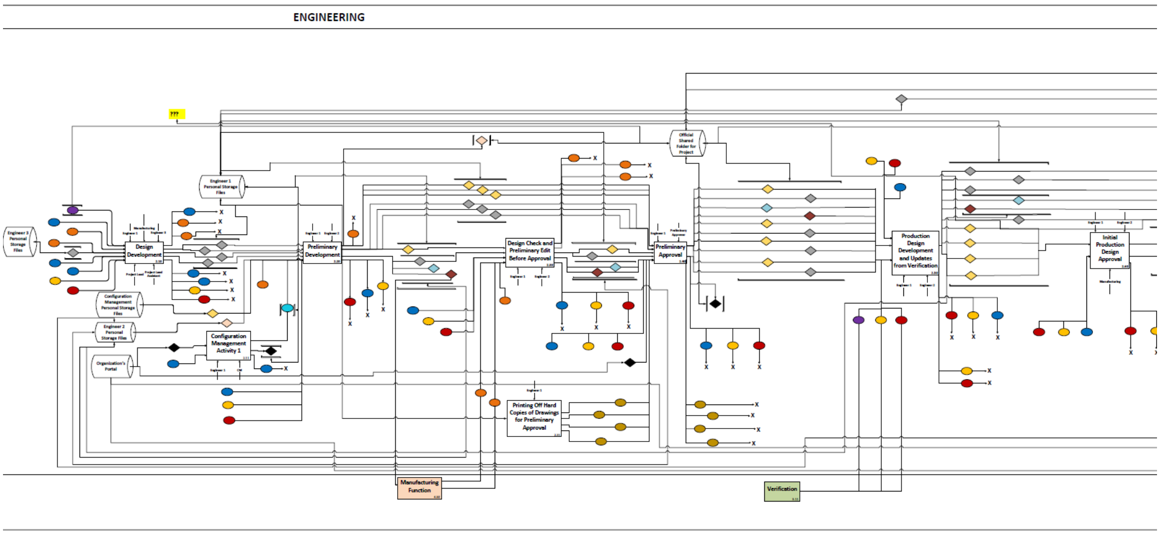


Figure 6.12: Engineering Color-Coded Data Vessel Mapping (Part 1)

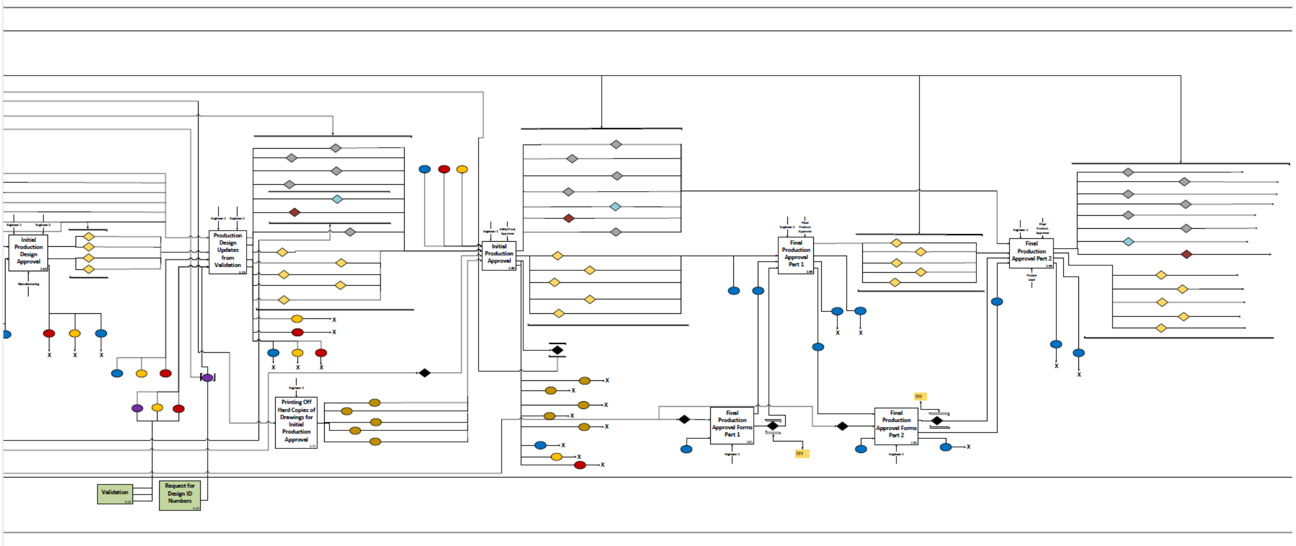


Figure 6.13: Engineering Color-Coded Data Vessel Mapping (Part 2)

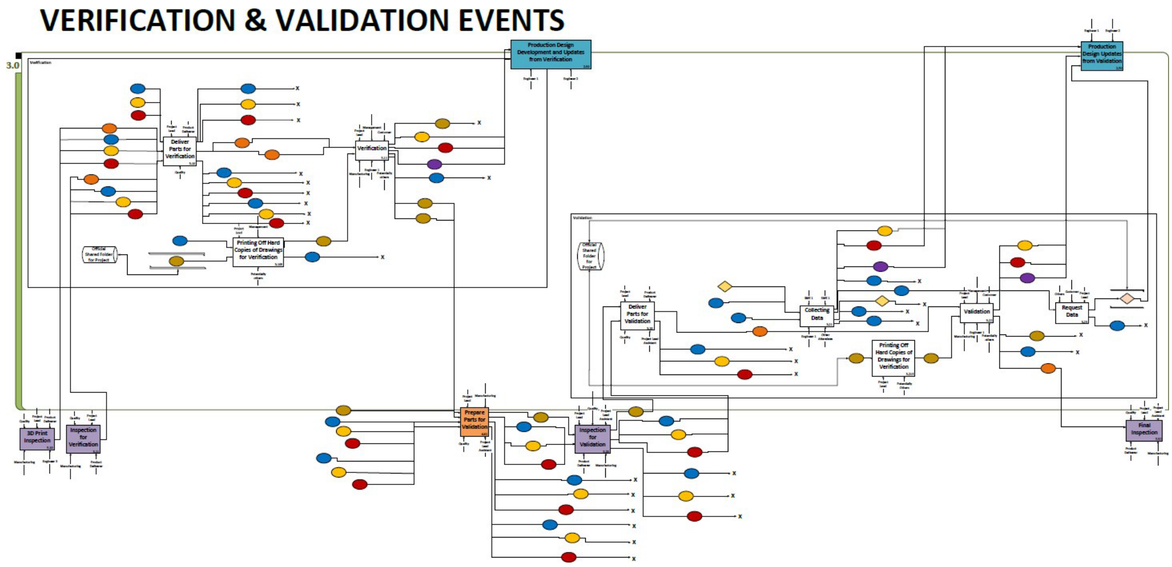


Figure 6.14: Verification & Validation Color-Coded Data Vessel Mapping

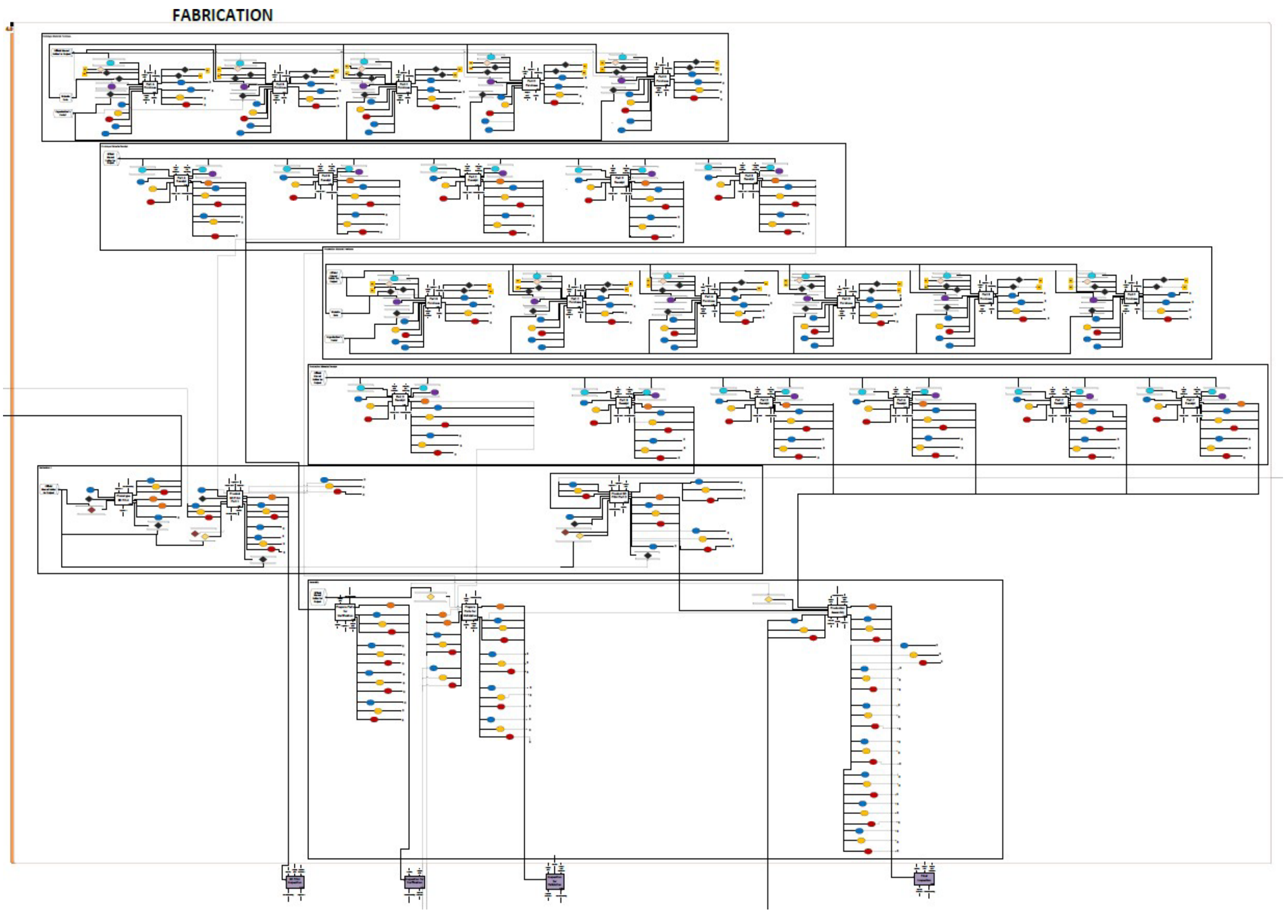


Figure 6.15: Fabrication Color-Coded Data Vessel Mapping

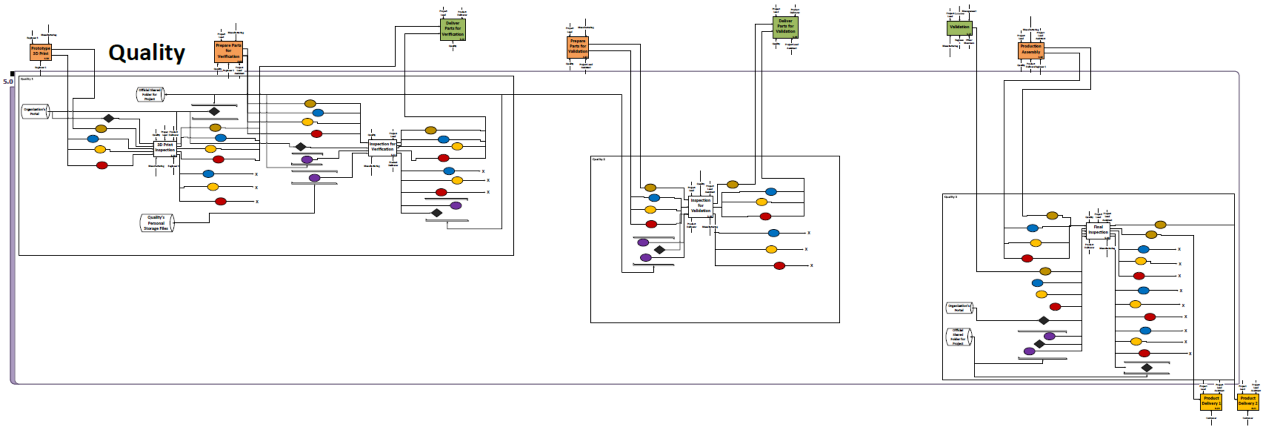


Figure 6.16: Quality Color-Coded Data Vessel Mapping

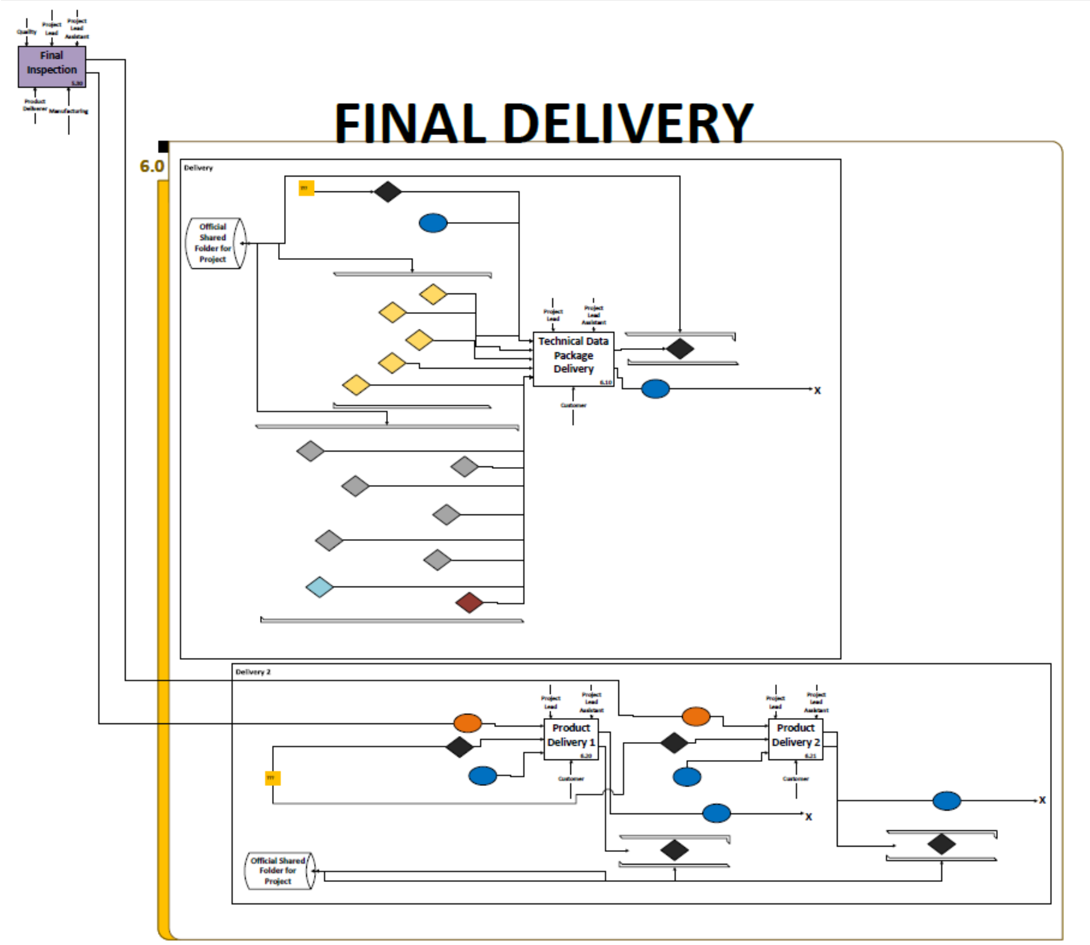


Figure 6.17: Final Delivery Color-Coded Data Vessel Mapping

6.1.3 Step 3: Data Element Level Mapping and Analysis

To achieve the Data Element Level View, the Data Vessel Level Mappings were used to locate and investigate data vessels to capture the data elements contained in the vessels. From there, the flow of data elements was listed using the flow captured in the maps from Step 2. The Data Vessel Level View revealed approximately 1,000 data vessel inputs and outputs across the functional activities with more than 500 data vessel dependencies. The Data Element Level View revealed over 2,500 unique data elements in the 25,000 total data element instances. The Engineering Functional Area produced the largest amount of data elements, with around 1,500 unique Engineering Data elements and with just under 12,500 data element instances. Over 90% of the data element instances were manually exchanged, therefore, less than 10% of data element instances were digitally connected.

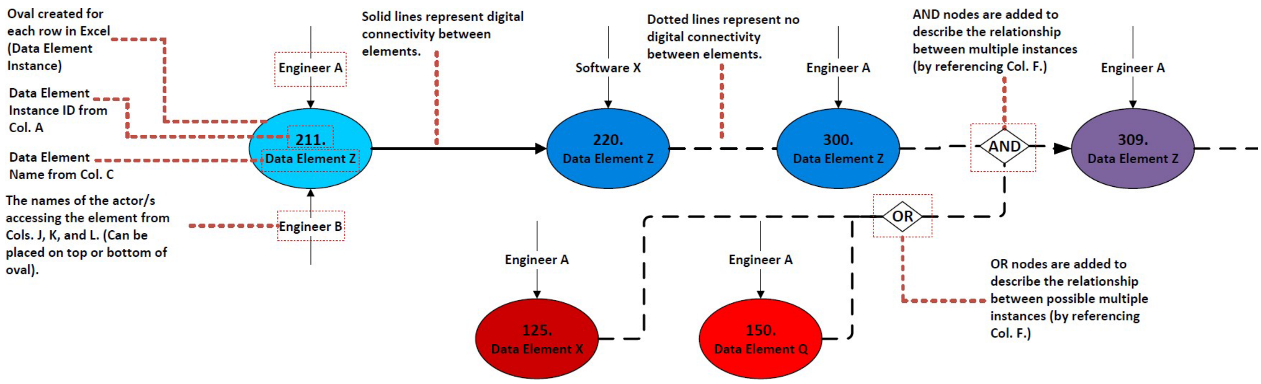


Figure 6.18: Data Element Mapping Syntax

Figure 6.18 shows the syntax of visual mapping of the flow of data elements from the Data Element Level View Listing. Figure 6.19 shows the data element thread of the nomenclature of an additively manufactured part. The color of the ovals are darkened to represent the maturation of data element across the data element instances. The white ovals represent data elements relationally connected to the nomenclature of the additively manufacturing part but that are different than the nomenclature of the additive part (ex. part dimensions, part due date, etc.). Altogether, this thread was made of around 157 data element instances, but for the sake of simplicity, only a segment from that data thread is shown in Figure 6.19.

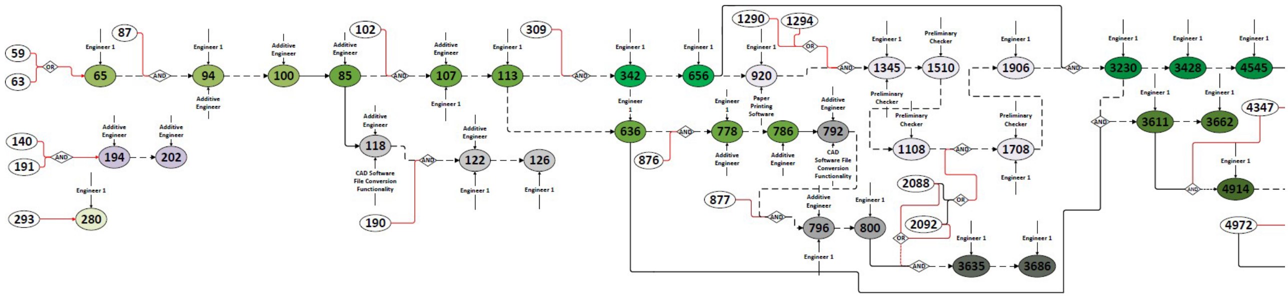


Figure 6.19: Current State of an Isolated Data Element Thread of AM Part Nomenclature

This map is made by creating an oval for each row in Excel and labeling it with both the instance ID and the data element name. The color of the oval darkens as the data element matures across the product realization process. Arrows are then added to either the top and/or the bottom of the oval with the actor(s) accessing the data element in that instance. Connecting lines are added to describe the flow and digital connectivity of the data elements.

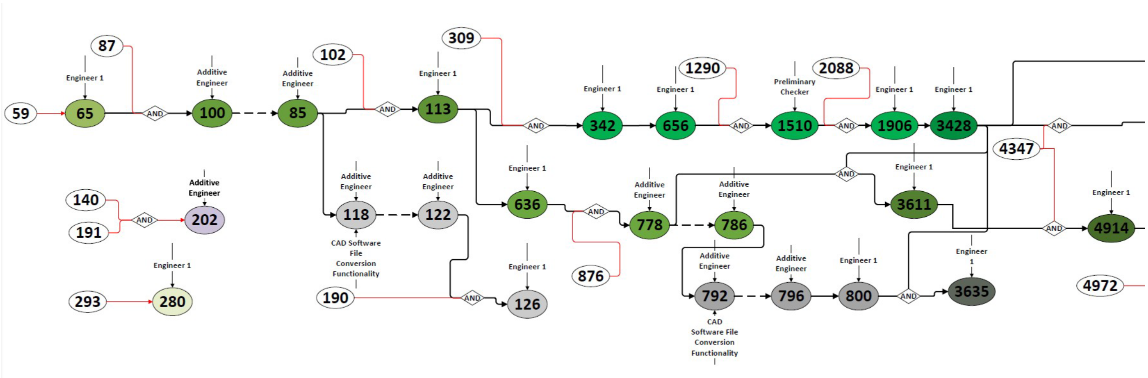


Figure 6.20: Improved State of an Isolated Data Element Thread of AM Part Nomenclature

A relationship between the current data element and past data elements is established by referring to the "Linked to Instance ID" in Col. F. This column presents the previous data element instance or instances in which the current data element is directly linked. If the current data element is directly linked to multiple data element instances, "AND" and "OR" logic, or some combination of the two, are used to describe the relationship between the multiple instances. The "AND" logic indicates that all of the data element instances directly preceding the current data element instance are relationally linked to the current data element instance. The "OR" logic indicates that in reality, only one of the directly preceding data elements instances are linked to the current data element and there is no standardization as to which one it is.

After determining that a relationship exists between the current data element instance and the data element referenced in Col. F, it must be determined whether the relationship involves digital connectivity or not. Some form of digital connectivity occurs if at least one actor in the data element instance is a software. A lack of digital connectivity is described by a dotted line connecting the data elements, and digital connectivity is described using a solid line.

Figure 6.19 presents the data thread shown in Figure 6.20 and how it would be improved if a digital technology such as a PLM system were used to digitally connect the elements when possible. It was found that reorganizing the flow of instances for this data element and connecting them with digital technology could reduce the total number of data elements in that thread from 157 to 111 (a 29.1% reduction). Digital connectivity could be increased from 22.3% to 75.4%. Therefore, DEMA was proven to be an essential tool in enabling digitalization.

According to the works of Sztipanovitz et al. [64], the most critical kinds of design knowledge for reuse are that which relates to system models and testing/verification methods. It is interesting that this application of DEMA revealed that data related to verification and validation was the most likely to be unstructured (5.3% structured) and the least likely to be saved for reuse (2.9% saved for reuse).

The example data element thread shown in Figures 6.19 and 6.20 showed that through the application of DEMA, digital connectivity could be increased from 22.3% to 75.4% for an additive manufacturing data element. Additive manufacturing is an inherently digital process, and it is likely that other data elements start with less connectivity than this example's 22.3%. Therefore, the results of this study make sense considering the findings in [82], that a transition from a drawing-based process to a model-based process would reduce cycle time by 74.8%. More research is needed to confirm the improvement metrics.

6.2 Results from DEMA Application in a Modeling and Simulation Operational Environment

In this application, DEMA was utilized to enable enhanced system definition for the development of a verification and validation process of a modeling and simulation operational environment. The results revealed that DEMA is a practical tool for both improving existing systems and defining new systems, and the data element level-view captured by DEMA can be used to define the interconnections between the system elements that will be inputs to SysML models. One of the main differences between this application of DEMA and the first application, was that this application was focused on enabling Model-Based Systems Engineering (MBSE). MBSE is different from the traditional, document-centric system engineering approach in that it utilizes system models to capture work products and understanding across the system lifecycle [126]. The traditional document-based approach exacerbates system failures caused by a lack of mechanisms for enforcing consistency between SE artifacts and communications across the lifecycle [126].

Based on the review of literature, it was determined that DEMA can be used to enable Model-Based Systems Engineering by defining the interconnections between the system elements inputs within the SysML models. Therefore, MBSE models can serve as the mechanism to demonstrate connections and deploy SE artifacts across the lifecycle. DEMA provides a method to identify and define the data elements and interconnections that will go into the models so that MBSE models can be created and deployed across the lifecycle in a systematic, timely, and verifiable manner.

A systematic literature review (SLR) was conducted by Sandia National Laboratories to search for industry case studies that could support the business case for MBSE implementation [127]. The results of this review found over 88 studies that reported improvements from an MBSE approach across various use cases, such as defense, space, and commercial systems [127]. Another SLR by Henderson and Salado captured 360 papers with an aggregated count of 1,233 individually stated benefits from MBSE [128].

However, despite the potential benefits of MBSE, it has not been generally adopted [126], and this is at least partially because past MBSE publications have focused on anecdotal benefits rather than empirical proof of improvements [128][127][129]. Another reason may be that the benefits of MBSE are often portrayed in a way that overshadows its real-world barriers to entry [129]. Therefore, more research is needed that will lead to the development of tools, methods, and quantifiable business cases to enable the adoption of MBSE and reduce potential pitfalls.

SysML is a standard language for implementing Model-Based Systems Engineering [101]. There are various diagrams in SysML to create system models that define and show the interconnections between elements of a system [102]. Systems Modeling Language (SysML) can serve as the mechanism for connecting the Digital Thread and creating the Model-Based Enterprise, but by itself, it does not provide methods for systematically capturing hidden processes and data and information flows. Therefore, system analysis techniques are needed to accomplish this so that SysML and other modeling languages can be used to create connected models that accurately reflect the functions and data flows of real-world systems, and DEMA was used in this application to accomplish this goal.

Another of the main differences between the first application of DEMA and the second application was that this application focused on the verification and validation process of a modeling and simulation environment instead of a prototyping product realization process.

Verification and Validation (V&V) are two essential systems engineering processes in the development of systems [26]. Therefore, this application was more heavily focused on systems engineering. Although verification and validation may be similar in nature, their objectives are fundamentally different [16]. As defined in IEEE Standard 15288-2008, verification is “a set of activities that compares a system or system element against the required characteristics” and may include requirements, design, and the system itself [18]. Validation, however, is “a set of activities ensuring and gaining confidence that a system” can meet stakeholder requirements in the “intended operational environment” [18].

Verification and validation in various applications are governed by standards including but not limited to: AS9100 for aerospace quality [16], MIL-STD 3022 for United States Department of Defense verification, validation, and accreditation of models and simulations [130], and IEEE Standard 1012-2026 for system, software, and hardware verification and validation. Given the diversity and number of potential applications and standards for verification and validation, it is reasonable to assume that confusion may arise as to what verification and validation entails. Another potential area for confusion comes from the fact that verification and validation are not simply one-time events.

For example, IEEE STD 1220-2005 *Systems Engineering — Application and Management of the Systems Engineering Process* defines the main steps of the systems engineering process as being requirements analysis, requirements validation, functional analysis, functional verification, synthesis, design verification, systems analysis, and control with various tools and methods prescribed at each step [17]. The recommended practices guide for MIL-STD 3022 also outlines multiple verification and validation activities in their framework, including conceptual model validation, verification, data verification and validation, and results validation [131]. Therefore, this application also explored utilizing DEMA to uncover terminology disconnects surrounding verification, validation, and system development.

Despite the complexities and potential pitfalls of verification and validation, the literature is clear that V&V is necessary for timely completion and success in system development, as a lack of verification and validation can result in increases in cost and schedule as a result of latent defects causing defective materials and rework [113]. In terms of MBSE, research has shown that the use of authoritative system models can theoretically be used to reduce lifecycle cost and enable enhanced verification and validation [113][127].

As digitalization enables more complex and software-centric systems, there is an increasing need for rigorous verification and validation of critical systems [113]. NASA estimates that the cost of verifying the software of complex systems may cost 75 to 88 percent of the total cost for developing the software [132][113]. Research conducted by Campo et. al found that one of the most positively perceived attributes of Model-Based Systems Engineering to be its verification and validation capability, but they also found that many of claims made about MBSE in the papers identified were based on author opinions and not supported by metrics [129]. Therefore, more research is needed into methods that can be used to quantitatively define the impact of MBSE on system verification and validation, and this DEMA application was conducted to enable this.

6.2.1 Step 1: Functional Level Mapping and Analysis

The Functional Level View analysis defined the system functional architecture of the verification and validation (V&S) process of the modeling and simulation (M&S) environment. In DEMA, the Functional Level View analysis is iterative, and in this application, the stakeholders were in the process of developing the V&V system, so the definition of functional areas and activities required multiple round-table discussions with system stakeholders to ensure the architecture met system requirements. In the end, 4 functional areas, 15 sub-functional areas, and 79 functional activities were defined. The functional areas were defined first: Management of Environment Development, Integration, and V&V; Development of Environment Components; Integration of Environment Components; and V&V of the Environment (Figure 6.21).

The discussion then continued with stakeholders to determine what functional activities would fall under each functional area. While doing this, several sub-functional areas were allocated to group-related functional activities. Within Management of Environment Development, Integration, and V&V, the sub-functional areas were 1.10 Project Management, 1.20 Management Meetings, and 1.30 Configuration Management. Within Development of Environment Components, the sub-functional areas were 2.10 Component Design, 2.20 Component Development Part 1, and 2.30 Component Development Part 2.

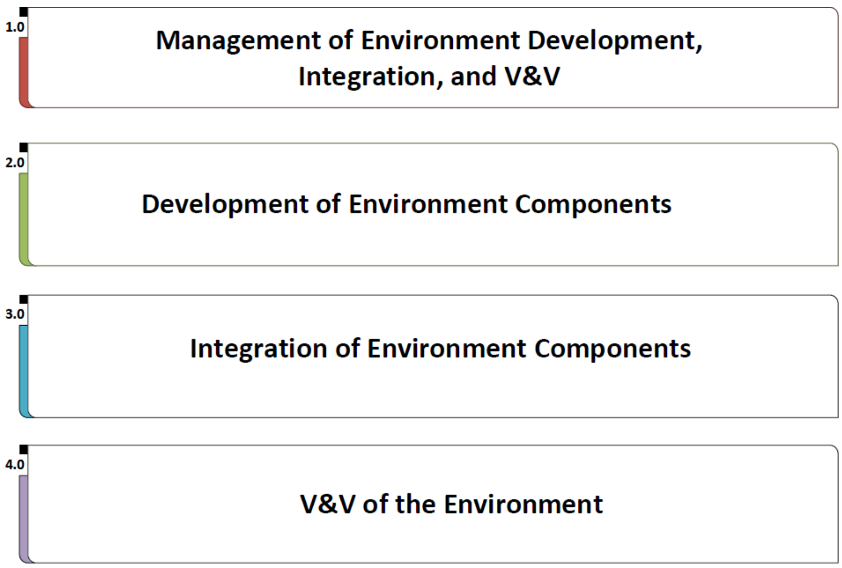


Figure 6.21: High-Level Functional Areas of the V&S Process of the Modeling and Simulation (M&S) Environment

Within Integration of Environment Components functional area, the sub-functional areas were 3.10 Pre-Integration and 3.20 Environment Integration. Within V&V of the Environment, the sub-functional areas were 4.10 Initial V&V Planning, 4.20 V&V of Environment Designs, 4.30 Environment Validation Planning, 4.40 Environment Verification Planning, 4.50 V&V Scheduling, 4.60 Creation of Verification Plan, and 4.70 Environment V&V Execution. The visual mapping of the sub-functional areas within the functional areas is shown in Figure 6.22.

The 79 functional activities were then determined (Figure 6.23). The activities are labeled with numbers instead of names to protect the privacy of the organization that owns the system. Although this DEMA functional mapping was made for a specific verification and validation process of a modeling and simulation environment, the functional architecture was created with an agnostic approach. This enabled the organization to use and edit the architecture as needed for reuse in the future development of verification and validation processes for operational environments.

During round table discussions, notes were also captured describing the nature of individual activities. During the system development, the system terminology was created. Therefore, whenever the stakeholders elucidated on the nature of an activity, their thoughts were captured and detailed descriptions of each activity and the identification of the actors being interviewed that were participating in those activities were systematically captured in DEMA Step 2.

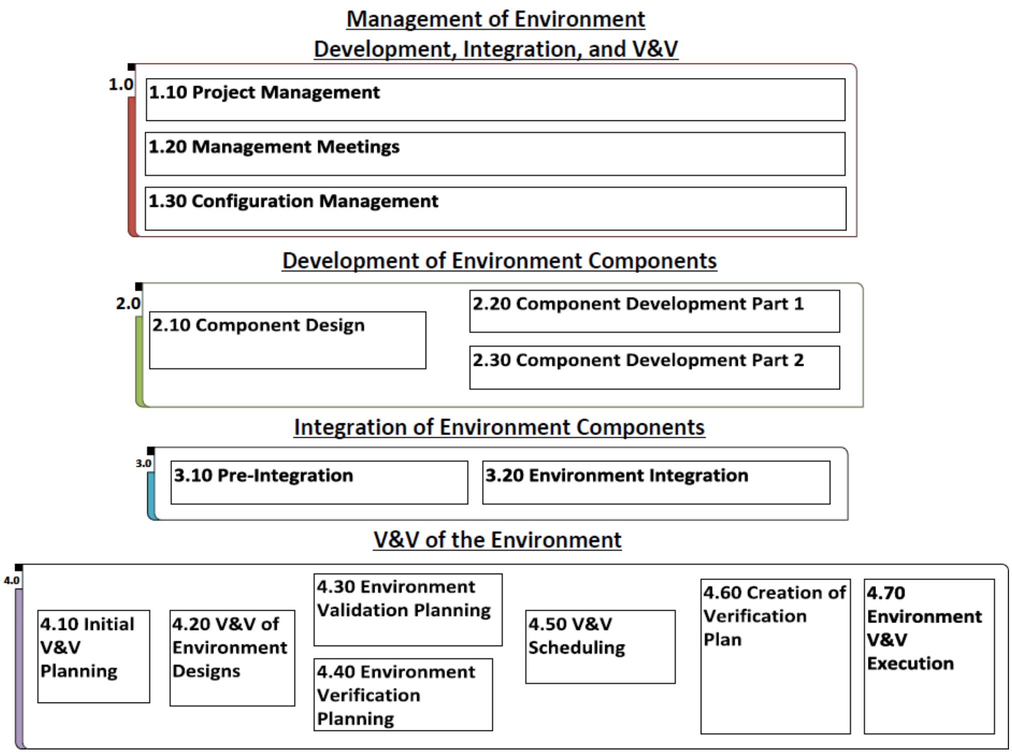


Figure 6.22: Sub-Functional Areas of the V&S Process of the Modeling and Simulation (M&S) Environment

6.2.2 Step 2: Data Vessel Level Mapping and Analysis

The Data Vessel Level View was created by conducting standardized interviews with the actors (from engineering and management) who were developing, using, and managing the system. There were six interviewees: the environment software developer, the project lead representing Organization A, the project lead representing Organization B, the operational coordinator, and a subject matter expert with knowledge of the systems that would later be using the completed simulation environment.

The interviews began by showing the participants the completed Functional Level View Mapping (Figure 6.23) and asking them to identify which of the 79 activities with which they were involved. Multiple interviewees were involved in several of the same activities, but because of limited stakeholder resources, each activity was only covered once by an participant except in the very beginning of the interviews, where a few activities were covered by multiple participants. Whenever multiple participants were associated with an activity, the participant interviewed for that activity was based solely on which participant was available first for an interview. Alternative mappings were made where the participants had different responses for activities. The ability to have different stakeholders address different functions speaks to the practicality of the DEMA method because real-world system stakeholders have limited time and resources. Ideally, all system stakeholders would be interviewed for all activities in which they are involved; however, that would be a costly approach to employ.

During the interviews, four of the functional activities, 1.31, 1.33, 1.34, and 2.33, were found to have no owners, meaning that none of the system actors were associated with the activities and the activities were not being performed. The data vessel mappings revealed the consequences of the four activities having no owners. For example, the effect of three of the Configuration Management activities (1.31, 1.33, and 1.34) having no owners is that the 75 other non-configuration management activities are not governed by acceptable configuration management controls or methodologies.

If the Configuration Management activities had owners and were being performed, there would be an exchange of data vessels with the other 75 activities. These data vessels could potentially include operating procedures that would show the actors how to conduct configuration management requirements for each activity. Activity 2.33 is a function where procurement activities occur, and the consequence of this activity having no activity owner is that there is no visibility into how procurement data flows into Functional Area 3 activities where the procured items are integrated into the system.

Past system analysis techniques such as Value Steam Mapping, IDEF0, Business Process Modeling Notation, and SysML diagrams do not offer a way to systematically uncover hidden activities while enabling a view of individual data elements. Therefore, the DEMA methodology is novel in its ability to uncover hidden activities and data elements and threads, and it aggregates tribal knowledge that reveals opportunities for improving the system. For the activities that did have owners, the interviewee's responses were recorded in the interview templates.

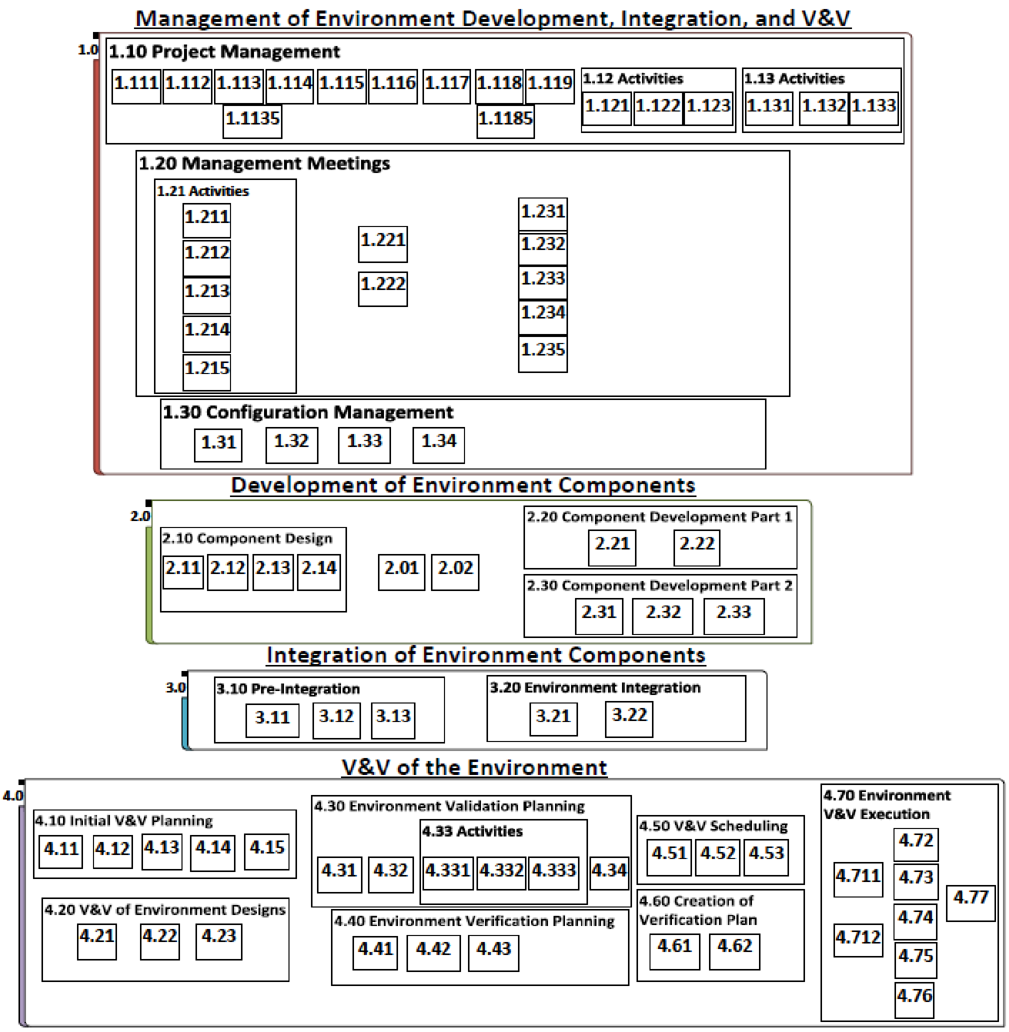


Figure 6.23: Functional Activities of the V&S Process of the Modeling and Simulation (M&S) Environment

The participant was asked for each of their activities to provide a high-level summary of the activity; to identify which personnel were involved with the activity; to identify if any V&V decisions were made in the activity and if so, what data vessels were associated with those decisions; to provide estimates for the minimum, max, and most likely calendar time associated with the activity; to identify the directly proceeding and succeeding activities; and to identify the data vessel inputs and outputs.

If the activity was iterative in nature, the decision criteria to exit the activity was also recorded. If the activity is iterative in nature and has clear exit criteria that must be met, this is also recorded next to the name of each data vessel in brackets ”[]”. The place of storage, personnel who handled the data, data format, tools used to handle the data, personnel the data vessel was sent to, and means of transfer were also recorded for the data vessels. The standardized interview template used to record the data vessel information and an example of a completed questionnaire for a generic functional activity are provided in the appendix.

Once the interviews were completed, the results were mapped using a technique derived from an integration of IDEF0 and DeMarco’s Data Flow Diagrams to create the Data Vessel Level View Mappings. The syntax for this mapping is shown in Chapter 3 in Figure 5.4. The data vessel inputs are listed on the left side of the function box that identifies the activities, and the data vessel outputs are listed coming out of the right side of the box. The personnel who were involved in the activity are listed on arrows on the top and bottom of the function boxes.

The cylindrical boxes represent the places of storage the data vessels were retrieved from and stored. If the data vessels could come from different places of storage “OR” nodes are used to indicate this condition. In this application of DEMA, iterations in functional activities were captured, and therefore the Data Vessel Mapping syntax was modified to accommodate this by having the data vessels that iterate back into the activity are connected by arrows back into the left side of the function box. The condition(s) for exiting the iterative activity are written in brackets “[]” next to the data vessels. The data vessel outputs that are connected to an “X” indicate that the data vessel was not stored anywhere for reuse.

If participants described different ways of accomplishing the same functional activity, then a separate mapping is created for each interviewee that reveals a potential need for standardization in how that activity is conducted. Each of the activities in the application were also mapped using this methodology. An example of a data vessel mapping for a generic activity is provided in the appendix.

An example of the mapping of one of the actual activities from Functional Area 1 is shown in Figure 6.24. Normally, the text describing the data vessel inputs and outputs is more descriptive than shown in Figure 6.24 but for the sake of the example they were shortened and generalized. Data vessel mappings were created for the functional activities in each of the four functional areas. Thirteen activities were left unaddressed by the organization due to the organization's time constraints.

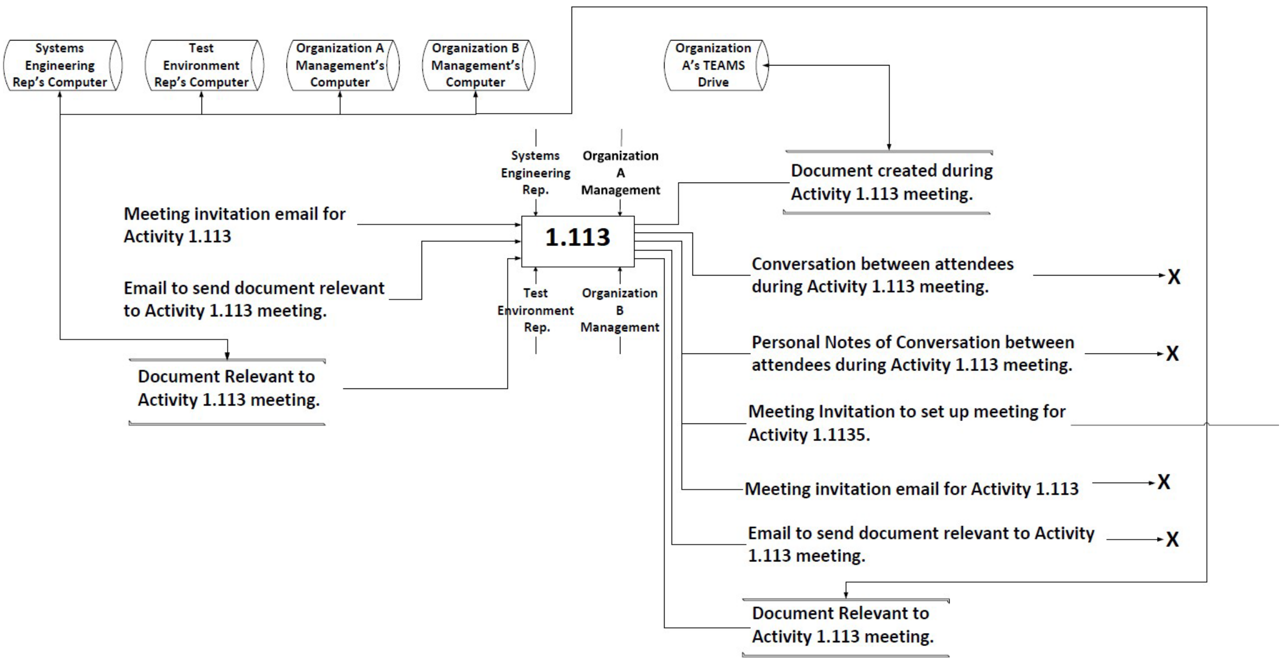


Figure 6.24: Data Vessel Mapping Activity 1.113

Over 500 data vessels were captured and mapped, and over 23 storage locations were identified as being used throughout the system to store these data vessels. Most of the data vessels were unique (different from one another). The high number of unique data vessels is from the high level of correspondence such as emails, conversations, and personal notes that had to occur to develop the system. The most reused vessel was the official V&V Document.

Once the mappings of the data vessels as they flow throughout the system architecture are documented, the data elements within each of the data vessels can be identified and documented. The flow of the data elements can be tracked, and individual threads of data isolated using the data element Excel sheet described in Section 3.3. From there, the data elements that are inputs into MBSE models can be identified, and an MBSE framework can be created that meets the requirements of the overall system architecture.

6.2.3 Step 3: Data Element Level Mapping and Analysis

Once the mappings of the data vessels as they flow throughout the system architecture are documented, the data elements within each of the data vessels can be identified and documented. The flow of the data elements can be tracked, and individual threads of data isolated, using the data element Excel sheet. From there, the data elements that are inputs into MBSE models can be identified and an MBSE framework can be created that meets the requirements of the overall system architecture.

The format for the Excel spread sheet is shown in Table 6.1. The Excel spread sheet format was modified slightly from the first DEMA application. A column was added to include the iterative criteria that determines when the functional activity is complete. Additionally, the first application included a column where the "Data Vessel Type" was recorded. During the second application, it was realized that the "Data Vessel Type" column was redundant in that the data vessel type could be determined from the "Data Format" column, so it was not included in the Excel sheet for the second application. Once a data element thread is isolated, the maturation of the element across the system lifecycle, how it changes formats, and the order in which system actors handle the element can be observed. Once the Data Element Level view is captured, the data elements to be implemented into the MBSE models can be identified and an MBSE architecture can be created.

6.2.4 Identification of Disconnects in System Terminology and Knowledge

As the Functional Level View and Data Vessel Level View interviews were being conducted, various terminology disconnects between the stakeholders were captured. When creating the Functional Level View, round-table discussions were held to define the functional architecture, and terminology disconnects were recorded by the DEMA analyst in their personal notes as they came up in the discussion. For example, there was debate among the stakeholders regarding where systems engineering would fit into the functional decomposition. Some suggested that systems engineering should have its own functional area, and others suggested that systems engineering was intrinsic to the activities within the functional areas. It was decided that systems engineering was intrinsic to the activities.

When creating the Data Vessel Level View, terminology disconnects were recorded in the standardized interview questionnaires and in personal notes. Some terminology disconnects were recorded explicitly in the questionnaire. As during the initial data vessel interview process, each participant was asked how they would define verification and validation to ensure that everyone was following the same understanding of the systems engineering terminology. The result of this discussion was the realization that many stakeholders were not on the same page about systems engineering terms and activities. This finding is consistent with the research of Kasser [133] and Rousseau [134] that show the need for unified systems engineering scientific principles, terminology, and activities. These results also demonstrate that DEMA can be used to uncover specific systems engineering terminology disconnects within an organization.

Other disconnects were captured in the interviewee's personal notes. For example, as each participant studied the visual mapping developed in the Functional Level View Analysis, questions arose regarding the nature of various activities and the DEMA analyst documented them. Altogether, there was confusion as to what approximately 30% of the functional activities entailed. The participants often initially struggled to understand that the functional analysis being conducted was specific to the development and V&V of an operational environment and not a specific system under test (SUT).

Overall, participants also struggled to understand the system view of the V&V

operational environment and that verification and validation must be conducted not only on the SUTs that would later utilize the environment but that the environment itself must be verified and validated. This is not surprising considering the review of literature that showed the opportunities for confusion surrounding V&V execution.

Table 6.1: Definition of Column Data in Data Element Level View Excel Sheet (Second Application)

Column Name	**Definition of Data**
Instance ID	A unique identifier for the data element instance.
Data Element ID	A unique identifier for the data element accessed in the instance.
Data Element	The name of the data element accessed in the instance.
Functional Activity	The functional activity under which the instance takes place.
Data Vessel Name	The name of the data vessel (as listed in the Data Vessel Mapping) in which the data element is accessed in that instance.
Iterative Criteria	The iterative criteria determine when the functional activity is complete.
Linked to Instance ID	The Instance ID of data element access instance that directly precedes the current data element instance.
Data Format	The format of the Data Vessel in which the data element is accessed in that instance (Text, PDF, Excel, Paper, Native CAD, etc.)
Place Where Data Resides	The place of storage where the Data Vessel in which the data element resides and is accessed in that instance.
Manual Handling	A "Yes" or "No" answer as to whether the transfer between the instance of data element access that directly preceded the current data element instance (referenced in the "Link to Instance ID" column) was manual (transferred between human actors). A "No" answer indicates that some form of digital connectivity exists between the instances.
New Data Element	A "Yes" or "No" answer as to whether the transition between the instance of data element access that directly preceded the data element instance (referenced in the "Link to Instance ID" column) introduces a new data element.
Actors 1, 2, and 3 (etc.)	The first, second, and third (etc.) person, software, or any other possible actor who is accessing and/or handling the data element in that instance.

Participants also used many different names to describe the operational environment. Some referred to it by the name of specific software used to conduct the operation. Others called it the "M&S", the "simulation", the "wrap around simulation", the "wrap around operational environment", and the "operational environment," often with a very ambiguous use of the word "model". There was also confusion about who performed what roles in the system. From a data vessel perspective, the participants used different words to mean the same thing for places of storage. For example, various team drives were referred to using different names, and the software developers were sometimes confused with other engineering roles.

Therefore, a better understanding of the terminology surrounding the system will facilitate practices happening more efficiently with fewer opportunities for mistakes in the system. The application of DEMA to the verification and validation process of the operational environment showed that DEMA is effective in uncovering terminology disconnects in systems engineering, functional execution, and data vessel usage. Once terminology disconnects are uncovered using DEMA, the system owner can determine a baseline for the terminology and create training materials and work procedures to govern system terminology and level the knowledge of the actors in the system.

6.2.5 Comparison of Applications

There were several other differences between the first DEMA application and this second application. Whereas the first DEMA application was applied to an existing system, this application was applied to enable the definition of a new system as it was being developed while simultaneously setting up the foundation for full traceability from the high-level system functions to the flow of data elements so that the elements could then be architected into an MBSE framework. Also, the Data Vessel Level View was modified in this application to capture iterations within functions and the decision criteria to exit iterative functions.

In the second application of DEMA it was shown that the methodology can be used to define new systems as well existing systems, such as shown in the first application. This application of DEMA provided comprehensive visibility into the definition of the verification and validation process of a modeling and simulation environment with four functional areas, 79 functional activities, and over 500 data vessel inputs and outputs identified and visually mapped. The results from this application were consistent with the previous DEMA application enabling the Digital Thread in that over 95% of data exchanges were undocumented and nonstandard, and over 50% of data vessel inputs and outputs were unstructured (in a format not ready for digital connection). Also, around 90% of data element exchanges were performed by the actors manually executing the exchange.

Various terminology disconnects were also uncovered between actors concerning V&V terms, places of storage, data vessels, activities, and roles. The presence of terminology disconnects revealed activities in which the actors in the system would waste time making mistakes due to incorrect interpretation of communications by other system actors. The DEMA application also revealed 23 disparate data vessel storage locations in the system as well as which data vessels were located in each place of storage and during which activity. Previously, this knowledge of the system was divided among the system actors and had never been combined, documented, and understood. Therefore, the DEMA results brought comprehensive visibility to the system whose function and data flows were previously undefined while also enabling a data element level view of the system.

Chapter 7

Quantification of DEMA Metrics of Improvement

To prove the efficacy of the DEMA approach in a quantifiable manner, the results from the application of DEMA in the product realization process are used. A sample of an improved data architecture was developed using the results from the first DEMA application to validate the DEMA methodology and to contribute to the business case for digitalization. Before this could be accomplished, a systematic approach had to be developed to take the initial DEMA results, identify sub-optimal data flows, and re-organize them to into an ideal state.

However, research has shown that fully enacting the digital thread can be both costly and time-consuming. One study found that creating robust Digital Thread and Digital Twin models for a U.S. Airforce aircraft would be comparable to the Manhattan Project in terms of the required resources and that the software development and sustainment for the Digital Thread aspects of the effort alone would cost anywhere from $80 to $180 billion [113].

Therefore, any method presented for creating an improved digital system architecture should prioritize identifying and improving the system's critical data threads based on data element reuse and potential metrics of improvement. This ensures that incremental steps can be taken towards realizing ideal data flows without requiring an organization to invest significant up-front resources to implement the Digital Thread and/or the Model-Based Enterprise. It was decided that a continuous improvement approach is ideal to be evaluated for taking incremental steps towards improved digital systems such as the Digital Thread and Model-Based Enterprise.

Continuous improvement, or kaizen, is one of the most important principles of the Toyota Production System (also known as lean manufacturing) [135]. Kaizen is a Japanese philosophy that focuses on continually improving operations and utilizing employee talent to make incremental changes to improve process efficiency [136]. Prior to Industry 4.0, continuous improvement efforts focused on improving the flow of physical products [112].

For example, Taiichi Ohno, the father of the Toyota Production System (TPS), developed the 7 Wastes of the TPS that established seven categories of non-value added activities that impede the efficient flow of physical materials in manufacturing systems. The 7 Wastes of the TPS are overproduction, waiting, transportation, over-processing, inventory, movement, and defects, and they are collectively used as a continuous improvement tool to identify and eliminate non-value added activities [112]. Value stream mapping is a visual mapping tool commonly used to identify wastes that can be eliminated in manufacturing systems [119].

Moving into the age of Industry 4.0 and digitalization, research is being conducted to apply lean manufacturing's continuous improvement principles to data and information flows. Yarbrough [112] evaluated past attempts to categorize data and information waste and then developed Taiichi Ohno's mental model and applied it to identify novel waste categories for data and information flows. Other research has utilized value stream mapping to eliminate waste in manufacturing information inefficiencies [76][77], but these efforts also do not address the data element level that DEMA enables and therefore do not support full digitalization.

Therefore, a new continuous improvement methodology is proposed for Digitalization that allows stakeholders to use the DEMA results with data and information waste identification categories and metrics of improvement to prioritize what improvements are chosen for initial implementation. This approach allows incremental steps to be taken toward implementing digital systems such as the Digital Thread and Model-Based Enterprise. This methodology integrates the DEMA methodology with the data and information waste categories developed by Yarbrough [112] and various systems engineering techniques.

7.1 Evaluation of Metrics of Improvement

Several metrics of improvement for data and information flows were evaluated to determine which would be addressed in this new continuous improvement application. Roh et al. [76] proposed five performance metrics for data and information flow efficiency: level of automation, centrality index, real-time capability index, media disruption index, and first past yield index. Figure 7.1 shows a table created by Yarbrough that provides the equations used to calculate each of Roh et al.'s metrics and provides an explanation for each [112].

The metrics presented by Roh et. al can holistically quantify the efficiency, configuration management risks, and time metrics of data and information flows, and therefore they were chosen to be evaluated as part of this continuous improvement methodology. However, as pointed out by Yarbrough [112], Roh et al.'s metrics do not provide a clear method for determining the Return on Investment (ROI) for using digital technologies to enable digitalization, and these metrics were not used to assess improved states of data and information flows. Therefore, Yarbrough presented metrics that are tied to specific data and information wastes and that can be converted to costs [112].

Yarbrough determined which of her eight waste categories for data and information flows were related to the data being 1) the right data (for the user to make decisions), 2) in the right place (of storage), 3) at the right time (for the user to make decisions), and 4) in the right form. Yarbrough presented various metrics for each of these four data characteristics, shown in figures of the tables created by Yarbrough in Figures 7.2 - 7.5 [112]. Metrics from Roh et al. [75] and Yarbrough [112] were chosen to be incorporated into the continuous improvement methodology.

7.2 A Continuous Improvement Methodology for Digitalization

A Continuous Improvement Methodology for Digitalization was developed to allow stakeholders to use metrics of improvement to prioritize the incremental steps that should be taken towards improving digital systems. According to the Department of Defense, Digital Engineering is the digitalization of the engineering and acquisition process by employing models in a Model-based Enterprise environment and the integration of technological innovations to enhance lifecycle activities [6].

Although Digital Engineering is necessary for creating robust, modern engineering practices, there is a significant need to develop various tools, processes, and methods for it to be enabled [6]. The continuous improvement methodology presented here enables the systematic reorganization and architecture of data and information flows into improved digital and model-based systems.

Performance Indicator Equation	Explanation
(1) Level of automation $$la = \frac{\Sigma i_a}{\Sigma i_a + \Sigma i_{na}} = \frac{\Sigma i_a}{i}$$ where, la equals the level of automation, i_a, an automized information transfers i_{an}, a non-automized information transfer. Note, that the sum of i_a and i_{na} is the total number of information transfers i.	"The *level of automation* is defined as the ratio of the number of fully automated information transfers to the total number of information transfers."
(2) Centrality index $$ci = \frac{\Sigma i_c}{i}$$ where, ci equals the centrality index, i_c, information transfers pointing to a central IT-system i, the total number of information transfers.	"The new method defines the *centrality index* as the quotient of the information transfers to a central IT system and the total number of information transfers."
(3) Real-time capability index $$rtci = 1 - \frac{\Sigma i_{nr}}{i}$$ where, $rtci$ equals the real-time capability index, i_{nr}, non-real-time capable information transfers i, the total number of information transfers.	"The *real-time capability index* is defined as the quotient of the number of information transfers in real-time divided by the total number of information transfers."
(4) Media disruption index $$mdi = \frac{\Sigma i_{d \to p} + \Sigma i_{o \to p}}{i}$$ where, mdi equals the media disruption index, $i_{d \to p}$, information transfers from digital to paper-based $i_{o \to p}$, information transfers from oral to paper-based i, the total number of information transfers.	"*Media disruption index* is defined, based on Ref. [23], as the sum of information transfers with a transition from a digital medium to a paper-based and from oral to a paper-based medium, divided by the total number of information transfers."
(5) First pass yield index $$fpy_i = 1 - \frac{\Sigma i_q}{i}$$ where, fpy_i equals the first pass yield for information, i_q, information transfers where a query is needed i, the total number of information transfers.	"The *first pass yield of information*, i.e. the query quota, is defined as one minus the quotient of the amount of information transfers for which a query is needed, and the total amount of information transfers."

Figure 7.1: Roh et al.'s Proposed Metrics for Data and Information Flows [76] (Figure of Table from [113]

		Related Waste Categories										
Waste Metric		Form	Excess	Error	Separation	Delay	Change	Manual Int.	Storage	Minimize (↓), Maximize (↑), or Control)	Definition	Unit
RIGHT DATA	Amount of Used Data		X						X	C	Data used by software or human	Gigabytes, Megabytes, or Kilobytes
RIGHT DATA	Amount of Unused Data		X						X	C	Data not used by software or human	Gigabytes, Megabytes, or Kilobytes
RIGHT DATA	Time Verifying/Validating	X		X	X	X	X	X		↓	Time spent evaluating data/information to check for accuracy and completeness	Time
RIGHT DATA	Time in Error Remediation	X		X		X	X	X		↓	Time spent correcting data/information errors	Time
RIGHT DATA	Percentage of Complete/Accurate Data			X				X		↑	Percentage of incoming data that is complete and accurate	Percentage

Figure 7.2: Yarbrough's Metrics for Data and Information Waste: Right Data (Figure of Table from [113])

		Related Waste Categories										
Waste Metric		Form	Excess	Error	Separation	Delay	Change	Manual Int.	Storage	Minimize (↓), Maximize (↑), or Control)	Definition	Unit
RIGHT PLACE	Count of Disparate Locations	X	X		X					↓	Number of disparate locations in which the same data/information (and its other forms) is stored	Count
RIGHT PLACE	Time Searching	X	X		X	X		X	X	↓	Time spent looking for the correct data/information	Time
RIGHT PLACE	Count of Transfers	X			X			X		↓	Transmissions of the data/information from the point of creation or collection to the point of use or storage	Count
RIGHT PLACE	Count of Instances of Missing Data	X		X	X					↓	Instances in which data/information is needed, but it is not present in its needed location	Count

Figure 7.3: Yarbrough's Metrics for Data and Information Waste: Right Place (Figure of Table from [113]

		Related Waste Categories										
Waste Metric		Form	Excess	Error	Separation	Delay	Change	Manual Int.	Storage	Minimize (↓), Maximize (↑), or Control)	Definition	Unit
RIGHT TIME	Information Lead Time				X	X		X		↓	Time it takes from the point of data/information creation or collection to the point of use or storage	Time
	Time Waiting				X	X		X		↓	Time spent waiting on data/information	Time
	Time at Rest		X			X			X	C	Time data is at rest or not being used	Time
	Time in Use				X	X		X		C	Time data is being used by software or human	Time

Figure 7.4: Yarbrough's Metrics for Data and Information Waste: Right Time (Figure of Table from [113]

		Related Waste Categories										
Waste Metric		Form	Excess	Error	Separation	Delay	Change	Manual Int.	Storage	Minimize (↓), Maximize (↑), or Control)	Definition	Unit
RIGHT FORM	Time in Manual Intervention	X				X	X	X		↓	Time spent to move data/information to the correct place or transform it into another form	Time
	Count of Manual Interventions	X		X	X			X		↓	Number of interventions that take place from the point of data/information creation or collection to the point of use or storage	Count
	Count of Forms	X	X					X		↓	Number of forms that data/information takes on from the point of data/information creation or collection to the point of use or storage	Count

Figure 7.5: Yarbrough's Metrics for Data and Information Waste: Right Form (Figure of Table from [113]

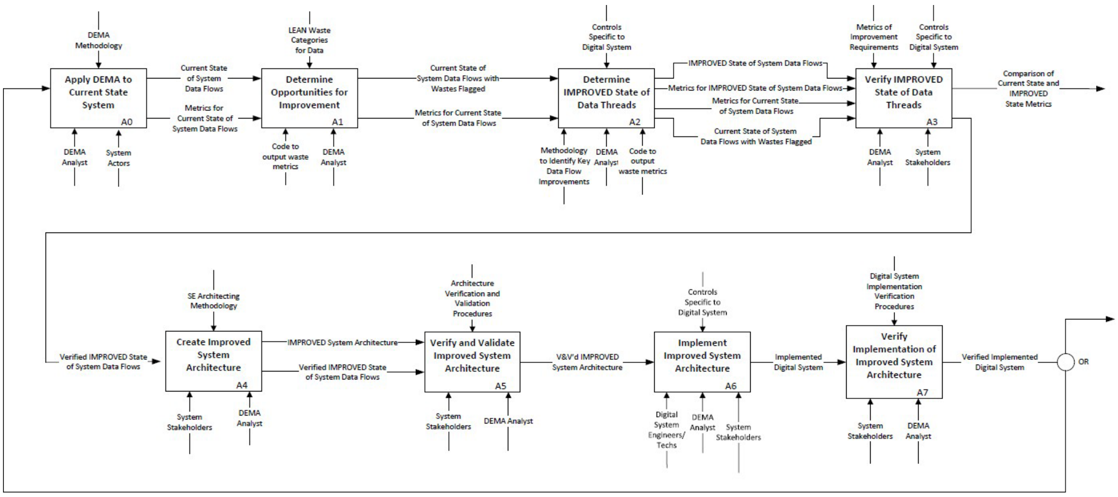


Figure 7.6: IDEF0 of a Continuous Improvement Methodology for Digital Engineering

The methodology was created by integrating the DEMA approach; Yarbrough's lean manufacturing data and information waste categorization [112]; and systems engineering architecture, verification, and validation methods. The DEMA methodology is based on a combination of traditionally siloed industrial engineering and systems engineering visual mapping techniques and analysis. Therefore, the development and application of this continuous improvement methodology presents an integration of industrial engineering, systems engineering, and lean manufacturing principles as a means of enabling Digital Engineering.

The continuous improvement methodology for Digital Engineering has 8 steps. The methodology is presented in the form of an IDEF0 in Figure 7.6, and the steps are as follows.

1. A0: Apply DEMA to Current State System - In this step, DEMA is applied to the system to be analyzed. It is constrained by the DEMA methodology and enabled by the DEMA analyst and system actors. The system to be analyzed is an input, and the outputs are the captured current state of the system's data flows (Functional, Data Vessel, and Data Element Level Views) and metrics for the current state of the system data flows (if available).

2. A1: Determine Opportunities for Improvement - In this step, the Data Element Level View Excel spread sheet is inputted into a code that identifies Yarbrough's [112] waste

categories within the current state data element flows and flags them within the data flows. The DEMA analyst conducts this step. The outputs of this step are the current state of data flows with flagged wastes and the current state metrics.

3. A2: Determine Improved State of Data Threads - In this step, the wastes flagged in the current state of data flows are used to determine what changes should be made to improve the data threads. The metrics of improvement, the controls specific to the digital system to be implemented (PLM, SysML, etc.), and a methodology to determine key data flow improvements are used to create the improved state of system data flows. The DEMA analyst conducts this step.

4. A3: Verify Improved State of Data Threads - In this step, the DEMA analyst presents the improved state and metrics of the system data flows to the system stakeholders. They compare the current and improved data flows to make sure that the improved metrics meet the system stakeholders' requirements for improvements (i.e., do the improvements provide enough cost savings, improved traceability, etc., to make the implementation of the improved state worth it?). The system stakeholders also review the improved state of data flows to verify that the improved state follows the controls of the digital system to be implemented (i.e., is the improved state feasible?).

5. A4: Create Improved System Architecture - In this step, the verified improved state of system data flows is used to create an architecture of the improved system. This architecture will be used directly to implement the improved system and should be created using systems engineering (SE) architecting methodology. The system stakeholders and DEMA analyst perform this step together.

6. A5: Verify and Validate Improved System Architecture - In this step, the improved system architecture is verified and validated by the system stakeholders and DEMA analyst. This ensures that the architecture has been created correctly (verification) and that the architecture accomplishes its intended purpose (validation).

7. A6: Implement Improved System Architecture - In this step, the verified and validated improved system architecture is implemented to create the digital system. The

engineers/technicians for the digital system (PLM, SysML) implement the architecture. The DEMA analyst and system stakeholders provide assistance for this step.

8. A7: Verify Implementation of Improved System Architecture - In this step, the system stakeholders and DEMA analyst verify that the improved system architecture has been implemented correctly. From there, the verified implemented digital system can be left alone, or the continuous improvement methodology can start back at A0 for further improvement.

7.3 Application of Continuous Improvement Methodology for Digital Engineering

The Continuous Improvement Methodology (CIM) for Digital Engineering (DE) was applied to a section of the data elements from the first application of DEMA that was conducted in the prototyping product realization environment. The data elements to which the CIM was applied were from the Engineering Functional Area. All steps of the continuous improvement methodology were applied to the Engineering data elements except for A6 and A7 because these steps involved the prototyping organization implementing and verifying the architecture and were, therefore, beyond the scope of this research. The prototyping organization is currently conducting steps A6 and A7, and future work may present those results.

Conducting Steps A0 - A5 were necessary to quantify DEMA metrics of improvement to validate the efficacy of the DEMA methodology, create a sample improved data architecture, and contribute to the business case for digitalization. This application of the Continuous Improvement Methodology for Digital Engineering also built upon the data and information waste categories and metrics proposed by Yarbrough [112] and Roh et al. [76] proposed metrics for data and information flows. The remainder of this sub-section presents the results and discussion of the application of steps A0 - A5.

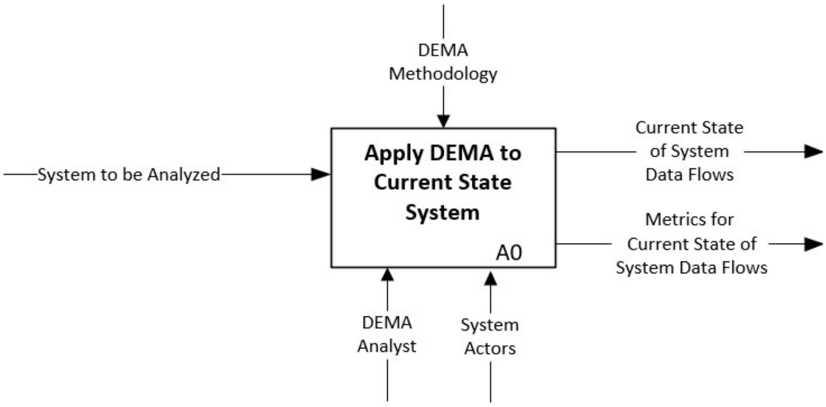


Figure 7.7: A0: Apply DEMA to Current State System

7.3.1 Step A0

In Step A0, DEMA is applied to the system that will undergo the CIM for DE. In this application, Step A0 encompassed the methodology and results presented in Section 5.2. Figure 7.7 shows an IDEF0 function block for A0. The A0 IDEF0 function is constrained by the DEMA methodology and enabled by the DEMA analyst and system actors. The system to be analyzed (in this case, the prototyping process) is an input, and the outputs are the captured current state of the system's data flows (Functional, Data Vessel, and Data Element Level Views) and metrics for the current state of the system data flows. For more information on Step A0 results, see Section 5.2.

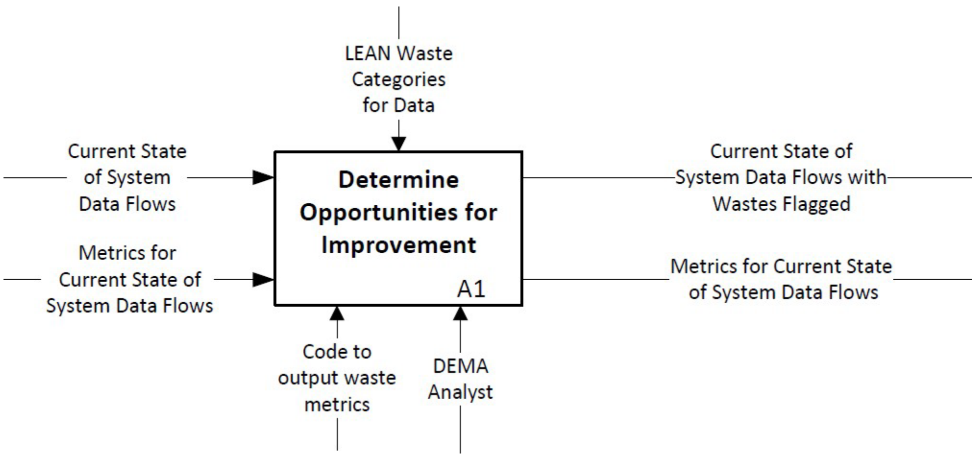


Figure 7.8: A1: Determine Opportunities for Improvement

7.3.2 Step A1

In Step A1, the opportunities for improvement to the current state of data flows were determined using a code that utilizes Yabrough's [112] lean waste categories for data and information. Figure 7.8 shows an IDEF0 function block for A1. The inputs to this step are the current state of system data flows captured during Step A0 and any metrics that are available for the current state. The outputs are the current state of system data flows with wastes identified (flagged) and the metrics for the current system of data flows. The DEMA analyst conducts this step.

- Define class "DataElementInstance"
  - #*This defines our class that we will write the information from the csv rows into.*
  - Define class attributes
    - Attribute to represent each of the column data from the csv file (see Figure 7.10 for column data)
    - Attribute that will store what rows (graph nodes) directly precede the current row (graph node)
    - Attribute that store flags for the wastes identified in each row (graph node) and row relationship (graph arc)
    - Attributes that store row relationship data (graph arcs)
  - Define class methods
    - Method to determine what wastes to flag in arcs and nodes (see Tables 7.2 and 7.3 to see how wastes flagged)
    - Method to determine what rows (graph nodes) directly precede the current row (node)
    - Methods to establish row relationships (graph arcs) with "AND" and "OR" logic preserved.
- Define main function
  - Iterate through row data from csv file and create class object instances for each row (graph node) using class attributes and methods.
    - Waste flags added as each row read in.
    - Various counters added for metric calculations (see Tables 7.3 and Tables 7.4 to see how metrics calculated)
  - Relationally link nodes and arcs to create graph structure after all row and row relationship data read into object instances.
  - Metrics calculated (see Tables 7.3 and Tables 7.4 to see how metrics calculated)
  - Print metrics to command line

Figure 7.9: Pseudocode for Step A1 Code

The pseudocode is shown in Figure 7.9, and a flowchart that shows how the code functions from a high-level perspective is shown in Figure 7.10. A detailed pseudocode is included in the Appendix. The code has defined a class ("DataElementInstance") that will represent each of the rows in the Data Element Excel Spread Sheet. This class has attributes that represent each of the columns in Excel as well as attributes that will be used to flag wastes and create the graph structure in a later part of the code. The first step is to read in the Data Element Excel Spread Sheet (.csv) as instances of the class. Figure 7.11 shows the format of the Data Element Excel Spread Sheet with example data. Chapter 5 explains what each of these columns represent in Table 5.3.

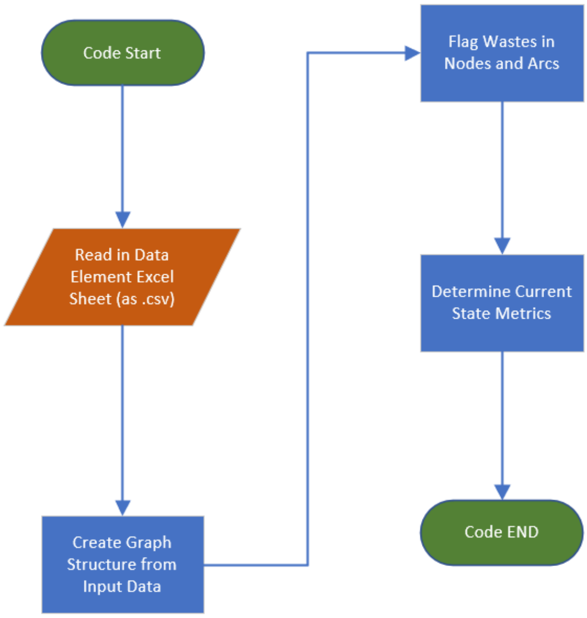


Figure 7.10: High-Level Code Flowchart

The prototyping organization provided an Integrated Master Schedule (IMS) that was used to assist in defining the activities for the functional level view in Step A0. When conducting Step A1, the timeline data from the IMS was used to estimate the labor hours associated with each data element instance (row). Column N, "Time Metric" was added to the Data Element Excel Spread Sheet. Each number in Column O represents an estimated working hour associated with that data element instance where the data was handled. Each data element instance within a function was assigned a time metric calculated by dividing the total working hours associated with a function by the total number of Data Element Instances in that function. Associating time metrics with each data element allowed for an estimate to be calculated for improvement by the reduction in labor hours from removing unnecessary data element instances.

Table 7.1 shows how the estimated time metrics for the elements in each function were calculated. Once the Data Element Excel Spread Sheet Rows have been read in as instances of the class, the code creates a directed graph structure where every data element instance is a node in the graph, and the arcs are determined by the "linked_relationships" attribute of each row read in from Column F of the Data Element Excel Spread Sheet (see Figure 7.11).

Column F includes conditional “OR” and “AND” statements to describe the relationships between directly preceding data element instances (explained in Figure 6.20), and these are captured in the graph structure as well. The graph structure is created and utilized in the code, but it is not currently printed. Figure 7.12 shows how the graph structure would be created for the thirteen example rows shown in Figure 7.11, with each node marked with the Instance ID from Column A.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Instance ID	Data Element ID	Data Element	Functional Activity	Data Vessel Name	Linked to Instance ID	Data Format	Storage	Actor 1	Actor 2	Actor 3	Manual Transfer Involved	New Data Element	Labor Metric (hr)
2	1	E1	Name of Part 5	Design Development 2.10	Relevant Document Needed for New Designs	N/A	PDF	Project Lead's Personal Email Directory	Project Lead	N/A	N/A	N/A	Yes	1
3	2	E2	Intro/Description of Relevant Document	Design Development 2.10	Email from Outside Organization with Relevant Document	N/A	Email Text	Project Lead's Personal Email Directory	Project Lead	N/A	N/A	N/A	Yes	1
4	3	E1	Name of Part 5	Design Development 2.10	Relevant Document Needed for New Designs	1	PDF	Shared Project Folder on Drive	Project Lead	Engineer 1	N/A	Yes	No	1
5	4	E3	Dimensions of the Off the Shelf Components of Part 5	Design Development 2.10	None	2	Observation	Internet	Engineer 1	N/A	N/A	Yes	Yes	1
6	5	E3	Dimensions of the Off the Shelf Components of Part 5	Design Development 2.10	Personal Notes taken by Engineer 1 of Dimensions of Part 5	1 and 2 and (3 or 4)	Paper	Physical Storage	Engineer 1	N/A	N/A	Yes	No	1
7	6	E4	Physical Location of Part 5 Component	TDP Development 2.20	Conversation about requirements for design models and where to access physical parts needed for design	5	Verbal/Auditory	Engineer 1's Memory	Engineer 1	Manufacturing	Project Lead	Yes	Yes	0.5
8	7	E4	Physical Location of Part 5 Component	TDP Development 2.20	Personal Notes taken by Engineer regarding conversation about requirements for design models and where to access physical parts needed for design	5	Paper	Physical Storage	Engineer 1	N/A	N/A	Yes	No	0.5
9	8	E5	Dimensions of Part that Affects Design of Part 5	TDP Development 2.20	None	6 or 7	Observation	Engineer 1's Memory	Engineer 1	N/A	N/A	Yes	Yes	0.5
10	9	E5	Dimensions of Part that Affects Design of Part 5	TDP Development 2.20	Personal Notes taken by Engineer 1 of Dimensions of Part 5	8	Paper	Physical Storage	Engineer 1	N/A	N/A	Yes	No	0.5
11	10	E6	Name of Part 3	TDP Development 2.20	Emails from Project Lead Assistant with Part 3 Info	8	Email Text	Engineer 1's Personal Email Directory	Engineer 1	Project Lead As	N/A	Yes	Yes	0.5
12	11	E7	Dimensions of Part 3	TDP Development 2.20	None	10	Observation	Internet	Engineer 1	N/A	N/A	Yes	Yes	0.5
13	12	E7	Dimensions of Part 3	TDP Development 2.20	Personal Notes takens by Engineer of Part 5	11	Paper	Physical Storage	Engineer 1	N/A	N/A	Yes	No	0.5
14	13	E4	Physical Location of Part 5 Component	TDP Development 2.20	Emails to Engineer 1 from Manufacturing and/or Project Lead about requirements for models and where to access physical parts needed for new designs	N/A	Email Text	Engineer 1's Personal Email Directory	Engineer 1	Manufacturing	Project Lead	N/A	Yes	0.5
15														

Figure 7.11: Example of Current State Data Element Excel Sheet for A1

Table 7.1: Calculation of Labor Hour Metrics for Data Elements in Engineering Functions

Engineering Function Working Hours	**2.10** 200	**2.20** 272	**2.30** 32	**2.40** 8	**2.50** 144	**2.60** 8	**2.70** 192	**2.80 - 2.90** 32
Total Data Element Instances	129	191	282	2,611	1,371	2,047	933	5,418
Labor Hrs. per Data Element Instances	1.550	1.424	0.113	0.003	0.105	0.004	0.206	0.006

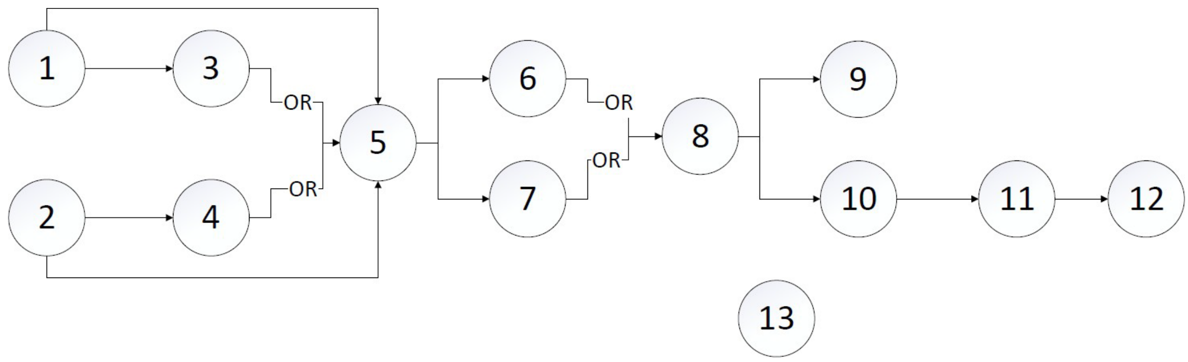


Figure 7.12: Example of Graph Structure from Rows in Figure 7.11

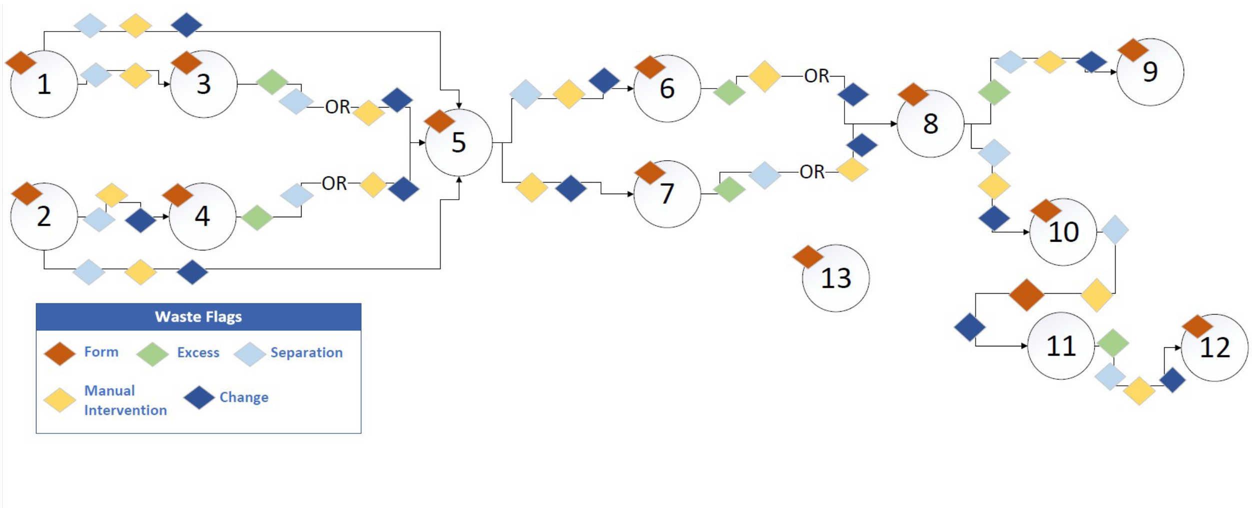


Figure 7.13: Example of Waste Flagged in Graph Structure from Rows in Figure 7.10

The next step of the code is to flag wastes in the nodes and arcs based on Yarbrough's [112] lean waste categories for data and information. Table 7.2 shows how each of Yarbrough's [112] were interpreted and flagged in the nodes and arcs. The "form" waste can be flagged directly from the node data, and "excess", "separation", "change", and "manual intervention" can be flagged based on the arc data. The DEMA analyst cannot determine when "storage" waste has occurred on their own because the storage of a data element instance that to the DEMA analyst appears to have no apparent purpose or requirement for preservation may have organizational requirements for preservation. Therefore, verification of the "storage" waste must be determined with the stakeholders in the organization.

"Delay" and "error" cannot be addressed given the current DEMA methodology. Delay cannot be addressed because it requires time measurement, and DEMA currently does not capture real time measurements. Error cannot be addressed because it would require knowing where the actor made mistakes and mistakes are currently not recorded in the data element instances. Figure 7.13 shows how the wastes would have been flagged for the example Data Element Excel Spread Sheet shown in Figure 7.11 using the methods for determining waste laid out in Table 7.2.

The only waste that would result in data element instances being eliminated is the waste of “excess.” Two types of excess waste were identified in the graph structure and are defined in Table 7.5. Ideally, all data elements flagged with of excess waste 1 can be eliminated. In each case where a data element instance is relationally linked to preceding instances with “OR” statements all but one of the preceding instances can be removed to eliminate excess waste 2. Form, separation, change, and manual intervention wastes are eliminated by changing the nodes object instances attributes. Currently, wastes are not eliminated by the code, but are strategically reduced in step A2 of the continuous improvement methodology.

The final step in the code is to determine the current state metrics based on the wastes flagged in the graph structure. The code was verified by running a smaller sample version of the Data Element Excel Sheet, manually calculating the metrics for the sample Data Element Excel Sheet and comparing the manually calculated metrics to the metrics output by the code.

Tables 7.3 and 7.4 define the current state metrics calculated in the code. Some of the metrics are calculated for individual data elements and other are calculated for the entire system (graph structure) and this is specified in Tables 7.3 and 7.4. Tables 7.5 and 7.6 present the results of the current state metrics calculated from the application in the prototyping organization. In these tables the term “unique” is used to describe a data element, data vessel, form, or place of storage, it means that the item is individually different from the other items.

7.3.3 Step A2

In Step A2, the improved state of the data threads is determined. The IDEF0 function block for A2 is shown in Figure 7.14. To accomplish this, an approach to identify key data flow improvements is used. The controls specific to the digital system to be implemented by constrain this step. For example, the prototyping organization was going to implement a Product Lifecycle Management (PLM) System as the means for connecting the data elements. Therefore, any proposed improved data threads have to be feasible within the PLM system.

Table 7.2: Application on Data and Information Waste Categories by Yarbrough [112]

Waste	Definition from [112]	Determination of Waste
Form	A format of data or information that is sub-optimal for use.	Determined directly from node data. If “data format” attribute (from Col. G of Data Element Excel Sheet) is unstructured, then the node is of sub-optimal form. If the node data is already structured, it does not necessarily imply that the type of structured data used is optimal. Identifying unstructured data is a step toward achieving more optimal formats. Shown as “◆” in Figure 7.13.
Excess	A greater amount or volume of data or information than needed.	Determined from by flagging arcs where the actors and data element name of the parent node are the same as the child node. Can also be determined from arc data if the arc is conditional (‘OR” statement). However, it cannot currently be determined directly from row data or arc data whether specific data elements are unnecessary for the project lifecycle. Shown as “◆” in Figure 7.13.
Error	Incorrect, inaccurate, or incomplete data or information.	Could be determined directly from node data, but not captured in the current state of DEMA. The opportunity for error can be indirectly evaluated based on arc data. If data is manually transferred, then an opportunity for data to be incorrect or inaccurate arises.
Separation	Data or information that lacks connectivity in its flow.	Can be determined directly from arc data. If the place of storage is different for the parent and child nodes, then the arc between the current node (child) and its parent node is separated. Shown as “◆” in Figure 7.13.
Delay	A stoppage in the flow of data or information.	Cannot be determined directly from node data or arc data. This is not captured in the current state of DEMA and would require time studies or real-time monitoring of the capture of data vessel/element handling.
Change	Manipulating, modifying, or transforming data or information.	Can be determined directly from arc data. If the data element instance changes from one form to another or from one data element to a different data element, change has occurred. Shown as “◆” in Figure 7.13.
Manual Intervention	Necessary intervention to initiate/ continue the flow of data or information.	Determined directly from arc data. If the node data “manual transfer” is “Yes” (Col. L of Data Element Excel Sheet), then the arc between the current node (child) and its parent node requires manual intervention. Shown as “◆” in Figure 7.13.
Storage	The retaining of data that has no apparent purpose or requirement for preservation.	Can be flagged, but not determined directly from the arc data. If a data element is used less than a certain number of times in the graph, then it is flagged as potentially having storage waste.. If a node is flagged as potentially having this waste, it must be evaluated by the system user to determine if the data does not need to be stored.

Table 7.3: Part 1: Current State Metrics for Data Elements Calculated by Code

Metric	**Calculation**
% of Form Waste	The total number of flagged FORM wastes divided by the total number of data element instances (nodes). Calculated for the system.
% of EXCESS Waste	For the system, it ishe total number of flagged EXCESS wastes divided by the total number of data element instance relationships (arcs). For individual data elements, it is calculated by the total flagged EXCESS wastes for a specific data element divided by the total number of data element instance relationships (arcs) for that data element.
% of CHANGE Waste	The total number of flagged CHANGE wastes divided by the total number of data element instance relationships (arcs). Calculated for the system.
% SEPARATION Waste	The total number of flagged SEPARATION wastes divided by the total number of data element instance relationships (arcs). Calculated for the system.
% of MANUAL INTERVENTION Waste	The total number of MANUAL INTERVENTION wastes divided by the total number of data element instance relationships (arcs). Calculated for the system.
% Potential STORAGE Waste	The total number of flagged potential STORAGE wastes divided by the total number of data element instances (nodes). Calculated for the system.
Estimated Labor Hours	The estimated labor hours for the total system is calculated by adding the labor hr. metrics for all nodes. The total number of labor hours for each element is calculated by adding the labor hr. metrics for each node of a data element instance.
Level of Automation (Roh et al. [76])	The total number of non-manual data element transfers divided by the total number of data element instance relationships (arcs). Also calculated for data elements (total non-manual data element transfers for one element divided by the total number of transfers for that one element).

Table 7.4: Part 2: Current State Metrics for Data Elements Calculated by Code

Metric	**Calculation**
Centrality Index (Roh et al. [76])	Calculated per place of storage. Defined as the quotient between the number of transfers to the place of storage and the number of total data element instance relationships (arcs). Places of storage include official places of storage such as shared drives and organizational portals, and unofficial places of storage such as actor's memory and email directories.
Media Disruption Index (Roh et al. [76])	The total number of transfers from digital form to paper form and from conversation (auditory) form to paper form divided by the total number of data element instance relationships (arcs). Calculated for the system.
First Pass Yield Index (Roh et al. [76])	One minus the quotient of the number of data element instances (for one element) divided by the total number of data element instance relationships (arcs). (Edited from Roh). Calculated for unique data elements.
Count of Disparate Locations (Yarbrough [112])	Count of unique places of storage used in the system and the number of times each storage place is used in data element instances (nodes).
Count of Transfers (Yarbrough [112])	Count of data element instances (nodes) for one unique data element.
Count of Manual Interventions (Yarbrough [112])	Count of data element instances (nodes) with manual transfer for each unique data element.
Count of Forms (Yarbrough [112])	Count of unique forms used in the system and the number of times each form is used in data element instances (nodes).
Actor Access Needs	The number of times that a data element or vessel changes actor access.

Table 7.5: Part 1: Prototyping Application Current State Metrics for Data Elements Calculated by Code

System Metric	Result
% of Form Waste	50.98%, with a minimum value as the ideal
% of EXCESS Waste 1 (where the actors and data element name of the parent node are the same as the child node)	27.56%, with a minimum value as the ideal
% of EXCESS Waste 2 (where the arc is conditional (”OR” statement))	13.72%, with a minimum value as the ideal
% of CHANGE Waste	65.24%, with a minimum value as the ideal
% of MANUAL INTERVENTION Waste	99.10%, with a minimum value as the ideal
% of SEPARATION Waste	66.88%, with a minimum value as the ideal
% Potential STORAGE Waste	Depends on the value specified by the user (storage value). For storage value = 5, the metric is 33.06%. For storage value = 10, the metric is 36.97%. A minimum value is the ideal.
Estimated Labor Hours	888.48 hrs., with a minimum value as the ideal
Level of Automation (Roh et al. [76])	0.90%, with a maximum value as the ideal

As shown in Figure 7.6, the Continuous Improvement Methodology for Digital Engineering is iterative and can be applied indefinitely to a system until all data and information flow inefficiencies are eliminated. The areas of the system that should be prioritized for improvement are determined by identifying the key data threads that have the most opportunity for improving the system metrics. Several of the current state metrics presented in Tables 7.5 and 7.6 are evaluated to determine the key data thread. The metrics used to determine the key threads are the excess wastes, estimated labor hours, level of automation, and actor access needs.

Table 7.6: Part 2: Prototyping Application Current State Metrics for Data Elements Calculated by Code

System Metric	Result
Centrality Index (Roh et al. [76])	Calculated per place of storage. The highest was 19.04% for the engineer's work computer. The lowest was 0.01%. A maximum value is ideal.
Media Disruption Index (Roh et al. [76])	74.86%
First Pass Yield Index (Roh et al. [76])	Calculated per unique data element. All data elements were over 99%.
Count of Disparate Locations (Yarbrough [112])	38 unique places of storage. The most used was the engineer's work computer with 3,397 instances (nodes). 11 storage locations had 25 or less instances (nodes).
Count of Transfers (Yarbrough [112])	12,982 for all engineering data element instances (nodes). The data elements with the highest count of transfers were related to the additive manufacturing data with up to 89 transfers.
Count of Manual Interventions (Yarbrough [112])	11,981 for the system. The data elements with the highest manual interventions were the dimensions of additive parts at 82, and the nomenclature of additive parts at 85.
Count of Forms (Yarbrough [112])	10 unique forms for the system. The most used form was paper at 3,116 data element instances (nodes). The least used form was a stl file at 108 data element instances (nodes).
Actor Access Needs	The data elements with the greatest actor access needs were additive manufacturing data with actor access needs around 48. 76.05% of the data elements had less than 5 actor access needs.

Table 7.7: Part 1: Prototyping Application Improved State Metrics for Data Elements Calculated by Code

System Metric	**Result**
% of Form Waste	55.62%
% of EXCESS Waste 1 (where the actors and data element name of the parent node are the same as the child node)	22.98%
% of EXCESS Waste 2 (where the arc is conditional ("OR" statement))	5.51%
% of CHANGE Waste	78.43%
% SEPARATION Waste	53.94%
% of MANUAL INTERVENTION Waste	74.34%
% Potential STORAGE Waste	Depends on the value specified by the user (storage value). For storage value = 5, the metric is 34.33%. For storage value = 10, the metric is 39.94%.
Estimated Labor Hours	660.78 hrs.
Level of Automation (Roh et al. [76])	25.66%

By evaluating the excess wastes for each data element, the DEMA analyst can determine what data threads have the most potential for unnecessary nodes and conditional arcs to be eliminated (thus reducing timely data handling and providing standardization in the data flows). By determining which data elements were associated with the greatest labor hours, the DEMA analyst can determine which elements may have the greatest return on investment to automate and control in the PLM system. By evaluating the level of automation for data elements, the DEMA analyst can determine which data elements have the most opportunities for improvement to be automated using the PLM system. Lastly, by evaluating the actor access needs, the DEMA analyst can determine which data threads are routinely needed by different actors across the lifecycle. By using the code to identify the data elements with the highest excess waste, labor hours, level of automation, and actor access needs, the DEMA analyst was able to determine that the key data threads for the prototyping organization were the data elements related to additive manufacturing data, part drawings and drawing approval forms, part revisions after validation, and drawing configuration management forms.

The DEMA analyst created an improved Data Element Excel Spread Sheet where these key data elements were automated in the PLM system. When creating the improved Data Element Excel Spread Sheet, the order in which the data elements flows to each actor is preserved and the excessive data element instances are removed when possible, based on the capability of the digital system being implemented. Whether or not a data element instance is excessive is determined based on the criteria associated with "excess waste 1" and "excess waste 2" in Table 7.3.

When a data element instance is linked to multiple preceding data element instances with "OR" logic, the DEMA analyst picks one of the multiple instances linked with "OR" logic to remain linked to the current data element instance. The other instances that were linked to the current data element instance are unlinked, and if they are not linked to any other instances, they are deleted. The order in which improvements are considered only makes a difference to the final outcome when determining which of the "OR" instances to remain linked to the current data element instance. In the current state of DEMA, this DEMA analyst decides this based on their knowledge of the system.

The updated Data Element Excel Spread Sheet was then run through the same code used in A1 to determine the metrics for the improved state of system data flows. The metrics from the improved state of system data flows from the prototyping application are shown below in Tables 7.7 and 7.8. When examining the improved state metrics for % of Form and % of Change, it appears that the metrics have become worse because they have increased. However, in reality, these metrics are only higher because the denominator of their equations (total number of data element instances) decreased because of excess wastes being eliminated.

Table 7.8: Part 2: Prototyping Application Improved State Metrics for Data Elements Calculated by Code

System Metric	**Result**
Centrality Index (Roh et al. [76])	Calculated per place of storage. The highest was 32.21% for the PLM system. The lowest was 0.01% for places of storage, including one of the actor's emails and some of the actors' mental memories.
Media Disruption Index (Roh et al. [76])	87.49%
First Pass Yield Index (Roh et al. [76])	Calculated per unique data element. All data elements were over 99%.
Count of Disparate Locations (Yarbrough [112])	34 unique places of storage. The most used was the PLM system with 4,968 instances (nodes).
Count of Transfers (Yarbrough [112])	10,231 for all engineering data element instances (nodes). The data elements with the highest count of transfers were related to the additive manufacturing data with up to 65 transfers.
Count of Manual Interventions (Yarbrough [112])	7,204 for the system. The data elements with the highest manual interventions were the dimensions of additive parts at 37 and the nomenclature of additive parts at 36.
Count of Forms (Yarbrough [112])	10 unique forms for the system. The most used form was paper at 2.342 data element instances (nodes). The least used form was a stl file at 78 data element instances (nodes).
Actor Access Needs	The data elements with the greatest actor access needs were additive manufacturing data with actor access needs around 37.

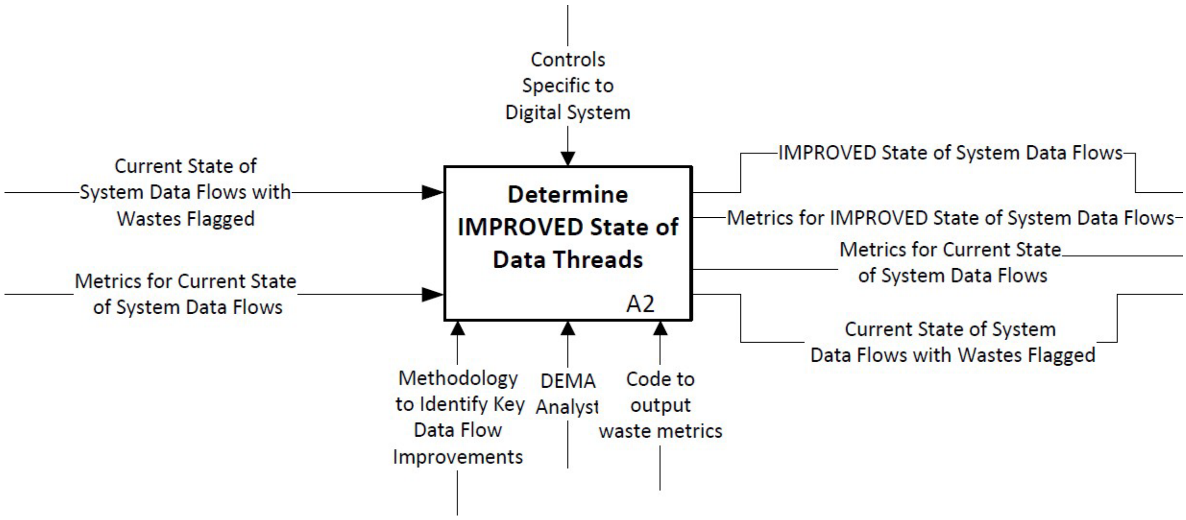


Figure 7.14: A2: Determine Improved State of Data Threads

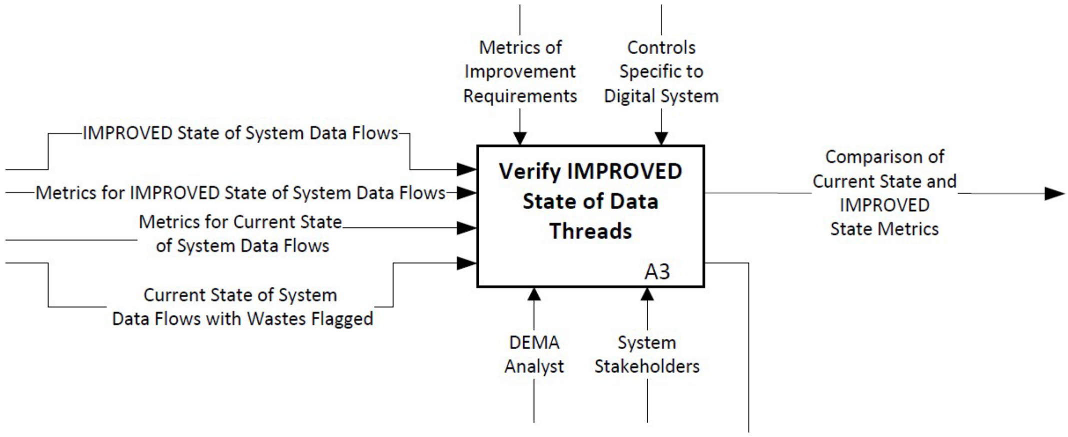


Figure 7.15: A3: Verify Improved State of Data Threads

7.3.4 Step A3

In Step A3, the verification of the improved data flows occurs. The IDEF0 function block for A3 is shown in Figure 7.15. The DEMA analysts presents the improved data flows and improved state metrics to the system stakeholders and asks the following verification questions:

1. Objective 1: Do the improved data flow follow rules of the controls of the digital system? (i.e., are the improved data flows possible to implement with the digital system?)
2. Objective 2: Are the improved data flows feasible? (i.e., does improved data flow require resources such as time and funding beyond the capacity of the stakeholders?)
3. Objective 3: If applicable, do the improved data flows include data that is prioritized by the stakeholders for reasons beyond the current state metrics?
4. Objective 4: Do the estimated metrics of improvement justify implementation?

Once the answer to each of the verification questions is affirmative, verification of the improved data flows is complete. Whenever an answer to one ore more of the verification questions is “no”, the DEMA analyst revisits Step A2.

7.3.5 Step A4

In Step A4, the verified improved data threads are used to develop an improved system architecture. The IDEF0 function block for A4 is shown in Figure 7.16. The DEMA analyst works with the system stakeholders to create the architecture to ensure that the architecture can be easily used by the system stakeholders for implementation in Step A6.

The architecture should be created using systems engineering architecting methods and created based on the requirements for the improved data flows. A section of the architecture created for the application with the prototyping organization is shown for the first engineering function in Figures 7.16 and 7.17.. Figure 7.16 shows the data vessel level view of the architecture, and Figure 7.17 shows the data element level view that exists underneath the data vessel level. Both of these architectures were created in Visio.

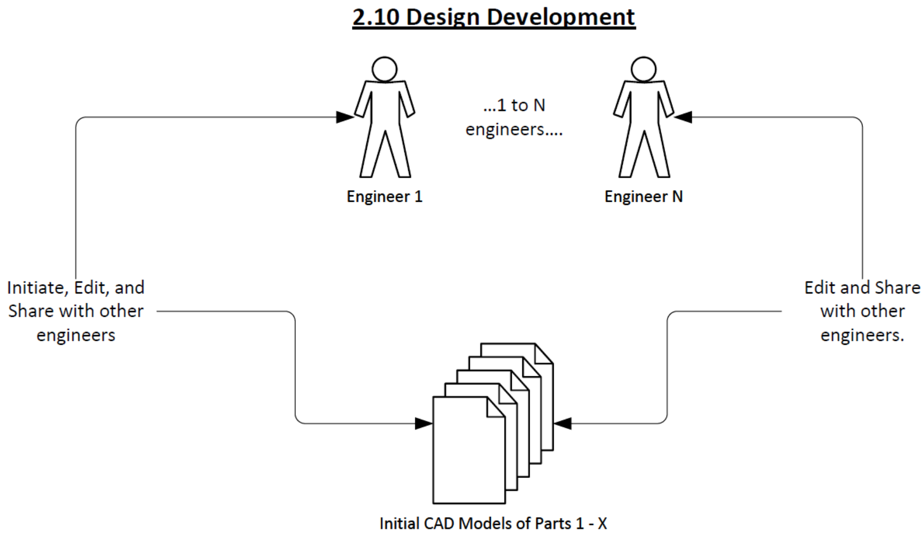


Figure 7.16: Data Vessel Architecture for First Function of Improved Engineering Data Flows

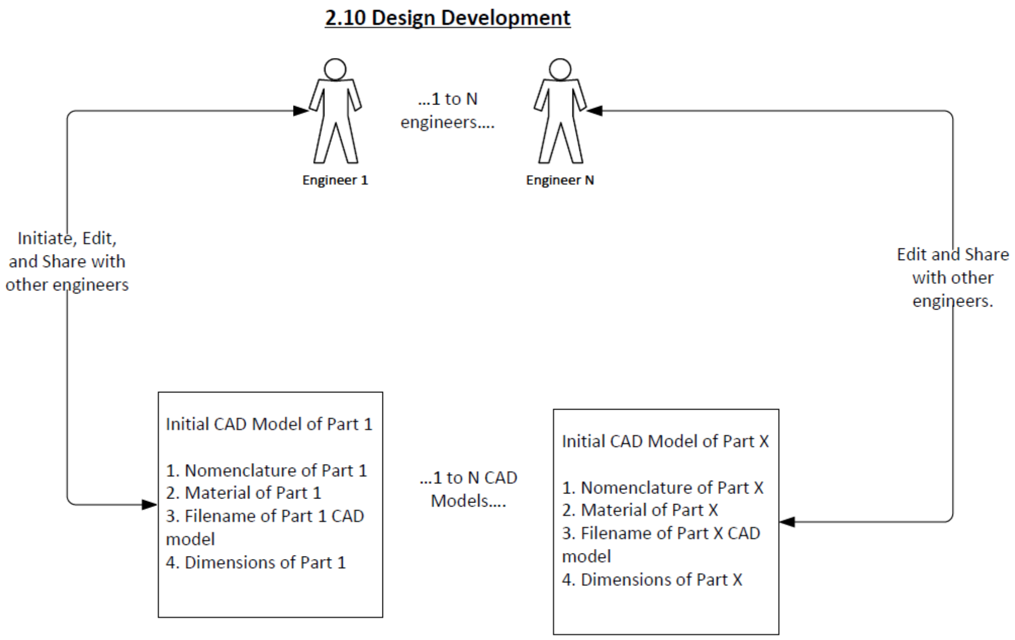


Figure 7.17: Data Element Architecture for First Function of Improved Engineering Data Flows

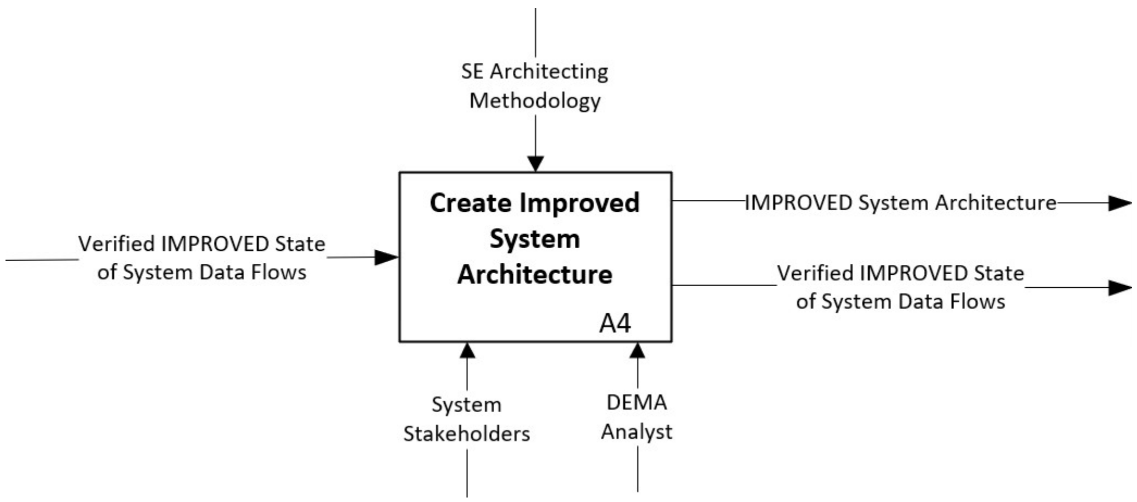


Figure 7.18: A4: Create Improved System Architecture

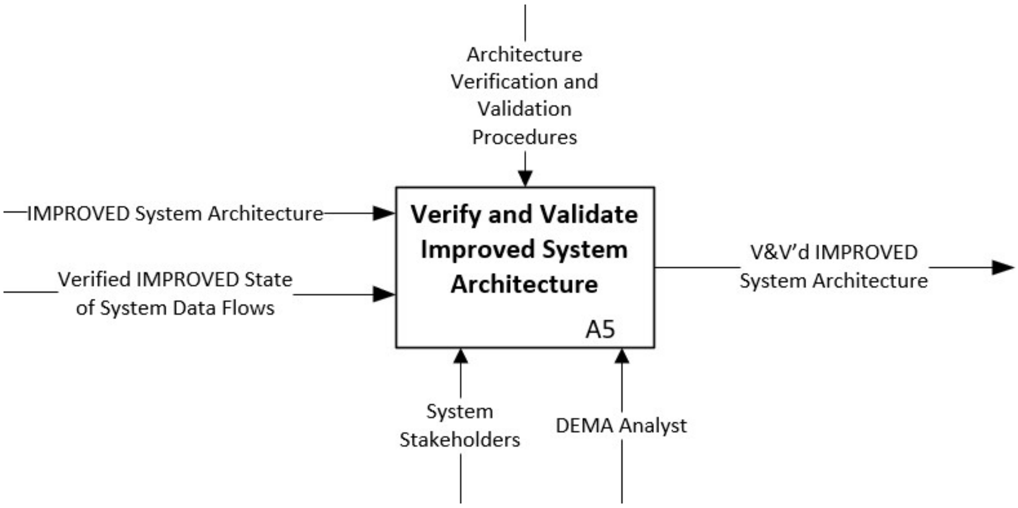


Figure 7.19: A5: Verify and Validate Improved System Architecture

7.3.6 Step A5

In Step A5, the improved state architecture is verified and validated so that it can implemented in Step A6. The IDEF0 function block for A5 is shown in Figure 7.18. The improved data flows were verified in Step A3, so the data flows do not have to be verified in this step again. The DEMA analysts presents the following questions to system stakeholders for verification and validation of the improved system architecture:

1. Verification: Is the improved system architecture correct? (i.e., does the improved system data flows achieve the requirements established for the system?)

2. Validation: Is the improved system architecture usable for implementation (easy to read, understand, includes all relevant architecture elements, etc.)?

Once the answer to each of these questions is affirmative, verification and validation of the improved system architecture is complete.

Chapter 8

Conclusion and Future Work

As the full impact of digital transformation continues to unfold [3][65], it is becoming increasing clear that digital transformation cannot be achieved by simply digitizing information or implementing stand-alone digital technologies [3][10]. Past visual mapping techniques are suited for digitization rather than optimizing the flow, form, and handling of data elements to enable digitalization. DEMA bridges this gap to allow for the realization of the Digital Thread and Model-Based Enterprise

DEMA is a novel approach for the standardized capture, mapping, and analysis of data and information flows to enable digitalization. Whereas past visual mapping has taken a functional, document, software-centric, and hierarchical views of a system, DEMA connects the system functions, data vessels, and data elements in three specific steps. The DEMA process also systematically uncovers the invaluable tribal knowledge of the people who work in the system and uncovers hidden and undocumented processes and data that make up a huge majority of the data exchanges in a system. DEMA integrates principles from industrial engineering, lean manufacturing, and systems engineering to create a novel continuous improvement tool which allows for the adoption of digitalization and can lead to the Digital Engineering environment desired by the DoD [6] and others.

This research showed that Data Element Mapping and Analysis can be used to enable enhanced system definition while capturing a data element level view that can be used for MBSE and Digital Thread architecture. This research also showed that the DEMA stakeholder elicitation pursued to the data element level revealed various terminology disconnects and misunderstandings related to the verification and validation of an operational environment along with undocumented sub-functions and activities. Therefore, the test and evaluation and systems engineering community at large can utilize DEMA to identify opportunities to reduce system engineering risks and failures.

To realize the advantages of Model-Based Systems Engineering, systems engineering tools are needed to move from a functional, document-centric view of data and information to a data element level view. This research demonstrated that DEMA is a new systems engineering tool that accomplishes this need. The results also showed that DEMA can be used for both improving existing systems and defining new systems, and the data element level-view can be used to define the interconnections between the system elements that will be inputs to SysML models or connected by the Digital Thread. Therefore, DEMA is a novel tool that can be used to enable MBSE, Digital Thread implementation, and enhanced system verification and validation.

MBSE models can serve as the mechanism to demonstrate connections and deploy SE artifacts across the lifecycle. DEMA offers the means by which the data elements and interconnections of models are identified and defined so that MBSE models can be created and deployed across the lifecycle in a systematic, timely, and verifiable manner. This research also showed that DEMA can be used to enable enhanced verification and validation of systems, and that by utilizing the current state maps at each of the DEMA steps, a kaizen approach can be utilized to incrementally connect the Digital Thread and eliminate inefficiencies in the system.

8.1 DEMA and Systems Engineering

DEMA can be applied to an existing system or in a system that is under development, as this research has shown. There are many ways in which DEMA can be leveraged in the systems engineering process. Within systems engineering, there is often a lack of integration between the involved disciplines, such as engineering, management, science, and finances [18], and DEMA can serve as an invaluable tool in defining the data relationship across the functional activities that correspond to these disciplines. DEMA can thus serve as an enabler of cross disciplinary system integration.

The definition of external system interfaces and their elements is one of the most important yet often overlooked systems engineering requirements tasks [137]. DEMA can be used in this context as a systematic method to uncover and define the data requirements for system interfaces for Systems Requirements Specifications. Graphical tools such as FFBD are encouraged for use when developing a Concept of Operations [138]. Therefore, DEMA can be integrated into the system engineering process by applying DEMA Step 1, the development of functional mappings for the Concept of Operations.

Within the concept of system science, there is the concept of "black box/white box" system representation. The "black box" view is that of the external system, and the "white box" is an internal view of the system that shows the structure of the elements [26]. An understanding of both the "black box" and "white box" view of the system and the relationship between the two is essential for system understanding [26]. DEMA is a tool to systematically determine and manage the data relationships between system elements and the functions to which they correspond, thus marrying the "black box" and "white box" views.

Within the Model-Based Enterprise, Model-Based Systems Engineering (MBSE) has captivated diverse sectors, including defense, commercial, and healthcare industries [39]. The three pillars of MBSE are modeling languages, modeling methods, and modeling tools [100]. There have been three significant periods in the evolution of data exchange standards that show a progression towards the utilization of system modeling language (SysML) and ontology-based data standards [63]. This evolution moves towards user-friendly modeling tools for exchanging data that have aspects of visual mapping techniques. Within the field of MBSE, SysML has emerged as the "de facto standard" [101]. A system model in SysML defines and shows the interconnections between elements of the system that represent key aspects, and there are various diagrams used within to SysML to accomplish this communication of system function [102].

SysML is both a language for data exchanges and a tool to create visual mappings, further proving the interdisciplinary nature of digitalization. Whereas SysML is primarily a language, DEMA is an approach, and SysML and DEMA can be used synergistically to enable digitalization in systems engineering efforts. SysML could potentially be used as a tool to visualize the mappings created during the DEMA steps, but SysML alone does not offer the DEMA methodology. DEMA could be used to identify the inputs that could be used with other

SysML diagrams. Therefore, DEMA and SysML can be used together to fully define a system model that accurately reflects the data and information of the system.

8.2 DEMA and the Principles of Lean Manufacturing

Past visual mapping techniques have relied on functional, document, and software-centric views of data and information. This is problematic given that the data elements, and not documents and software, are used to drive value in an organization. In the Toyota Production System, there is the concept that some activities in manufacturing are value-added, and others are non-value added, and the 7 Wastes of the Toyota Production System developed by Taiichi Ohno are used to identify the non-value added activities [14][139].

The data vessel is analogous to a physical container that is used to move and store materials in a manufacturing facility. In the same way a physical container is used for inventory management and transportation in manufacturing, the data vessel is used to facilitate the manual storage and transference of data elements across the project lifecycle. This reality hints at the possibility of wastes categories existing for data and information flows as Ohno's waste categories are for the analysis on physical processes, and research is actively being conducted to develop such categorizations [14][15][112]. Research has also been conducted in utilizing value stream mapping to eliminate waste in manufacturing information inefficiencies, but these efforts also do not address the data element level [76][77].

Value stream mapping, like other traditional mapping techniques, is suitable for functional and document-centric views of data and information flows as opposed to a data element level view that enables full digitalization. Therefore, DEMA offers a tool uniquely suitable for the application of lean manufacturing to data and information flows. Waste categorizations for data and information flows could be used to assist in identification and elimination of waste in both the data vessel and data element level views.

Two of the most important principles of the "Toyota way" are continuous improvement (kaizen) and respect for the people [135]. In most cases, it is not feasible to assume that all waste (non-value added activities) can be removed from a system, but continuous improvement allows a system to gradually be improved and optimized over time. This mindset is important to the application of DEMA, as the final mappings created by DEMA will uncover many sub-optimal and unstandardized data flows.

Although new technologies and the application of artificial intelligence and machine learning may eventually enable real-time optimization of data and information flows, no such technologies currently exist to enable this at the data element level. Therefore, a kaizen approach is necessary for the incremental improvement of a system. Future work will examine the application of DEMA to optimize data and information flows alongside the development of a system. The fact that the first step of DEMA creates a functional level view allows a system to be decomposed into discrete sub-systems that can be managed separately before being brought together for final data vessel and data element analysis. Ideally, DEMA would be conducted on a system in its entirety, but in some instances, this is not feasible because of system complexity or limited resources. Therefore, functional analysis can be used to incrementally apply DEMA to different functional areas of the system and thus enable continuous improvement that gradually connects the Digital Thread.

The term "Gemba" is the Japanese word for the real place in which work happens, and Taichii Ohno believed that the Gemba was the most valuable place to learn [15]. In DEMA, the Gemba is the fundamental enabler of the approach as all data and information flows are captured through interviews with the people who work in the system. This means that DEMA can be used as a novel approach to capture valuable tribal knowledge. The results of this DEMA application showed that data and information are not only siloed in disparate places of storage within computer systems but also within the minds of the people who work with the system.

One of the main principles of the Just-In-Time System is the elimination of "Overburden" or "Muda" on people and equipment [140]. Anecdotal data easily reveals the immensely negative impact of inefficient data and information flows [22] with the people in the system bridging gaps in processes with ad-hoc manual user interventions [64]. It is taken for granted that manual and costly data tasks are necessary, with little thought given to improvement [15].

The results of this research confirm these ideas and show that even well-performing systems rely on talented workers using siloed tribal knowledge to conduct heroic effort to manually handle data elements thousands of times across the life cycle. Such a system is not sustainable and relies on the overburdening of its people. Therefore, connecting the Digital Thread and eliminating waste in data and information flows is not just a matter of reducing costs and improving quality; it is an ethical imperative for the sake of removing overburden from the workers in the system.

8.3 DEMA and Digital Engineering

The definition of products is becoming exceptionally complex and 2D drawings are unsuitable for properly capturing this complexity [24]. Model-Based Definition has been proposed as an alternative to meet the needs of the modern enterprise. There is an ongoing effort to establish common information models as a means of enabling the Model-Based definition [13]. Persistent identification of product-definition elements was identified as a research gap in enabling Model-Based Manufacturing and Inspection [82]. DEMA can systematically reveal these elements and their relationships.

DEMA has been shown as a means of capturing all relevant data and information flows within a system and can be used alongside Model-Based Definition to realize the Model-Based Enterprise and establish complete product definition. Errors are more likely to accumulate as drawing based definitions are manually passed throughout the lifecycle [82]. Given that this research revealed thousands of data element instances related to the use of engineering drawings, this realization is even more impactful. Therefore, each instance of data element exchange and interaction is critical, and the realization of the Digital Thread and the Model-Based Enterprise would eliminate thousands of single-point failures in data elements.

8.4 DEMA and MBSE

The review of literature showed that past attempts to determine the necessary data for Model-Based Definition and the Model-Based Enterprise were narrative in nature and relied on methods such as surveys [13], workshops [24], and comparison studies [82]. This resulted in a significant research gap as no methods were discovered to holistically uncover the data and information flows of real-life systems down to the data element level view necessary for

Model-Based Definition.

In this application, DEMA was successfully applied to enable the definition of a new system as it was being developed while simultaneously setting up the foundation for full traceability from the high-level system functions to the flow of data elements so that the elements can then be architected into an MBSE framework. In the first application, Data Element Mapping and Analysis was proven successful in capturing the data element level in a well-defined process to enable the Digital Thread. Therefore, the DEMA methodology offers a novel approach by which the data elements and interconnections that will enable MBSE models and the Digital Thread can be identified and defined.

Some data elements in a system will not be repeatedly used throughout the system lifecycle or required for additional data vessel applications. For example, the date of a one-time organizational meeting may be only applicable to one functional activity, and thus the organization may choose to keep meeting dates in an email data vessel. Another example would be data vessel within a PDF form that the organization is required by a governing organization to maintain. Therefore, even an organization that has a fully developed and implemented an MBSE framework will likely have both MBSE data vessels and non-MBSE data vessels.

Research conducted by Campo et al. found that some of the negative aspects of MBSE were the perceived time commitment and high cost of implementation [129]. Therefore, a continuous improvement approach is recommended for MBSE implementation to reduce the cost and complexity of the MBSE system and to encourage a gradual move to MBSE rather than one that takes years for initial deployment. Because DEMA captures three levels of system complexity, continuous improvement is enabled at the Functional Level View, Data Vessel Level View, and Data Element Level View.

The Functional Level View can be analyzed to make sure that system requirements are met from a functional perspective, and adjustments can be made to the system architecture if necessary. Improvements can be identified at the Data Vessel Level View where short-term opportunities for improving information disconnects and unnecessary manual data handling can be identified. This identification is achieved by stakeholders reviewing the Data Vessel

Level Mappings and identifying areas for improvement.

For example, if an activity has correspondences (emails, conversations, and/or personal notes) that are not contained in official data vessels, then the reviewers could consider whether a new, standardized data vessel could be introduced that would allow the people in the system to be able to access the elements in the vessel for reuse throughout the development or sustainment lifecycle. This would eliminate opportunities for data elements to be lost, incorrect, and/or hard to find. Another example would be an activity that has data vessels being stored in disparate places (especially personal drives). Then the reviewers could create a standard procedure that the data vessels be stored in a specific location that is accessible to the relevant parties.

The Data Element Level View can be used to identify data that can be contained in MBSE models to eliminate information disconnects and manual handling. From there, an MBSE architecture can be created. Then, standards can be created that guide system personnel through the activities and data vessel handling (both MBSE and non-MBSE data vessels). This would be accomplished by documenting the terminology, ensuring common understanding, and guiding the user through the Functional Level Mapping and Data Vessel Level Mappings, after being updated with the improved data flows.

8.5 DEMA and A New Age for Verification and Validation and System Development

The application in the development of a new process showed that DEMA can be used to identify the knowledge and terminology disconnects related to the V&V of a system. Disconnects related to both systems engineering terminology, terminology for the specific system under development, and undocumented activities were identified. Therefore, the terminology disconnects, Functional Level Mappings and Data Vessel Level Mappings could be used as a foundation for creating training materials for the verification and validation of future systems.

The review of the literature found that one of the most positively perceived attributes of Model-Based Systems Engineering is its potential for enhanced verification and validation [129]. However, many of the claims made about the benefits of MBSE are based on author's opinions and not supported by metrics [129]. This application of DEMA enabled V&V to a new level of fidelity by capturing views of the system down to the data element level view.

Therefore, V&V activities no longer must be viewed as only discrete events with specific documentation. By using DEMA to capture the inner workings of a system down to the individual flows of data elements, verification and validation can happen throughout the entire lifecycle because the system is holistically captured, and V&V can be continuous.

MBSE can be used as the mechanism to manage and model the lifecycle data flows that will be used in the enhanced V&V. Also, the Hawthorne Effect shows that people who are subject to experiments or studies are inclined to modify their behavior because they are being observed [141]. Therefore, it is possible that the actors in a system where DEMA and MBSE have been used to enable full system visibility may be more inclined to conform to verification and validation guidelines.

8.6 Future Work

An agent-based model for information flows that modeled data interactions as physical counterparts such as fluid, momentum, velocity, force, and temperature as part of continuing work in the field of infodynamics was created by Waters and Ceruti [106]. Infodynamics is the field within information science of applying thermodynamic principles to information systems [142]. Future work could explore using principles of infodynamics to explore modeling DEMA results. This would enable the systems engineering, industrial engineering, and lean manufacturing principles of DEMA to converge with the field of information science.

Another research opportunity that would combine information science with industrial engineering and lean manufacturing principles would be to explore the application of determining the data and information flow equivalent of Therbligs. Therbligs were created by Frank and Dr. Lillian Gilbreth, and they are the fundamental motions that make up physical processes [143]. Data mining could be used to access and track the way actors interact with data on their computers. With data mining, the data and information flow equivalent of Therbligs could be developed with associated time metrics. This would enable the data and information wastes of delay and error (two of Yarbrough's [112] wastes that cannot be tracked in the current DEMA methodology) to be captured and quantified.

Other research has found that "representation" is the initial stage of realizing a digital twin [37][10], and DEMA could be used to assist in the creation of the representation stage of a digital twin. Other future work may include the utilization of machine learning or artificial intelligence in eliminating wastes in the data and information flows captured by DEMA. This work would also include outputting optimized future states of the flows. In its current state, DEMA is not a language for data exchanges, and work could be conducted to enable DEMA results to be outputted in a way that is readable to languages such as SysML. Broader future work could utilize and evolve the DEMA process to be suitable for analyzing cyber-physical systems, lifecycles for sustainability, cyber-security, healthcare systems, and supply chains.

Given that most MBSE literature has focused on anecdotal rather than quantifiable benefits of MBSE [128][127][129], the business case for implementing MBSE would be strengthened by the development of a method for calculating the empirical benefits of MBSE implementation. The third contribution of this research presented this in terms of the Digital Thread, but future work could do this for MBSE systems. DEMA captures the current state of a system down to a data element level view, and therefore if metrics can be associated with the reduction of data element instances and manual handling, then a quantifiable business case for MBSE could be made by comparing the data element metrics of a pre-MBSE and post-MBSE system.

The three pillars of MBSE are modeling languages, modeling methods, and modeling tools [100]. SysMl is a commonly used MBSE modeling language, and there are various modeling tools available for use, such as Cameo Systems Modeler, Rhapsody, and UModel [100]. A modeling method is a method for adding elements and defining the element's relationships into a system model, and common MBSE modeling methods include INCOSE Object-Oriented Systems Engineering and Weilkins System Modeling (SYSMOD) method [100].

Future work could investigate utilizing DEMA alongside modeling methods such as INCOSE Object-Oriented Systems Engineering and Weilkins System Modeling (SYSMOD) method to ensure that the created models accurately capture a system down to a data element level view [100]. Also, a systematic method needs to be developed to determine which data elements are the most critical to include in MBSE models. Machine learning techniques could be evaluated for such a method. The development of DEMA software that automates the interview process could be made to reduce the mental and time loads of the interviews in the future. Additionally, artificial intelligence could be used to determine what improvements need to be made in the data element flows captured by DEMA. For example, artificial intelligence could be used to determine which of the “OR” conditional data element instances are removed from the Data Element Excel Sheet instead of relying on the DEMA analyst to make the decision.

References

[1] H. Kagermann, N. A. of Science, Engineering, W. Wahlster, G. R. C. for Artificial Intelligence, J. Helbig, and D. P. AG, “Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative Industrie 4.0, final report of the Industrie 4.0 working group”, , 2013, Retrieved March 31, 2022.

[2] F. Galati and B. Bigliardi, “Industry 4.0: Emerging themes and future research avenues using a text mining approach,” *Computers in Industry*, vol. 109, pp. 100–113, 2019.

[3] T. M. Siebel, *Digital transformation: survive and thrive in an era of mass extinction*. Rosetta Books, 2019.

[4] V. Atluri, S. Rao, and S. Sahni, “The trillion-dollar opportunity for the industrial sector: How to extract full value from technology,” *McKinsey Digital*, 2018.

[5] O. Cann, “$100 trillion by 2025: the digital dividend for society and business,” *World Economic Forum*, 2018.

[6] M. Griffin, K. Baldwin, J. Stanley, R. Kewley Jr., and W. Bray, “Department of defense digital engineering strategy, june 2018.” Office of the Deputy Assistant Secretary of Defense for Systems Engineering, 2018.

[7] G. A. Harris, D. Abernathy, L. Lu, A. Hyre, and A. Vinel, “Bringing clarity to issues with adoption of digital manufacturing capabilities: an analysis of multiple independent studies,” *Journal of the Knowledge Economy*, 2021.

[8] D. Norquist, D. of Defense Office of Prepublication, and S. Review, “DoD digital modernization strategy, DoD information resource management strategic plan fy19-23,” 2019.

[9] C. Wittekind and G. Harris, “Searching for the inscrutable: A search for clarity in digital manufacturing definitions, terminologies, and technologies,” in *Institute of Industrial and Systems Engineers Annual Conference Proceedings, May 22–25, 2021*, 2021.

[10] A. Hyre, G. Harris, J. Osho, M. Pantelidakis, K. Mykoniatis, and J. Liu, “Digital twins: Representation, replication, reality, and relational (4rs),” *Manufacturing Letters*, vol. 31, pp. 20–23, 2022.

[11] J. Bloomberg, “Digitization, digitalization, and digital transformation: Confuse them at your peril,” *Forbes*, 2018.

[12] W. Holstein, “Industrial engineering,” 2017. Retrieved April 22, 2022, from https: //www.britannica.com/technology/industrial-engineering.

[13] S. Ruemler, K. Zimmerman, N. Hartman, T. Hedberg Jr., and A. Feeny, “Promoting model-based definition to establish a complete product definition,” *Journal of Manufacturing Science and Engineering*, vol. 139, 2017.

[14] A. Yarbrough, G. Harris, and G. Purdy, “Improving the flow of data and information in manufacturing,” *Manufacturing Letters*, vol. 32, pp. 1–4, 2021.

[15] A. Yarbrough, G. Harris, G. Purdy, and N. Loyd, “Developing Taiichi Ohno’s mental model for waste identification in nontraditional applications,” *ASME Open Journal of Engineering*, vol. 1, 2022.

[16] National Aeronautics and Space Administration (NASA), *NASA Systems Engineering Handbook*, rev 1 ed., 2007.

[17] Software and S.E.S.C. of the IEEE Computer Society, “IEEE std 1220-2005, IEEE standard for application and management of the systems engineering process,” tech. rep., ISO (International Organization for Standardization) and IEC (the International Electrotechnical Commission), 2005.

[18] Software and S.E.S.C. of the IEEE Computer Society, “IEEE std 15288-2008, systems and software engineering — system life cycle processes,” tech. rep., ISO (International Organization for Standardization) and IEC (the International Electrotechnical Commission), 2008.

[19] D. Kinard, “Digital thread and industry 4.0,” 2017.

[20] D. A. Kinard, “Digital thread and industry 4.0, NIST MBE conference.” Retrieved April 4, 2022 from https://www.nist.gov/system/files/documents/2018/04/09/2p_kinard_digitalthreadi4pt0.pdf, 2018.

[21] T. D. Hedberg Jr, *Enabling connections in the product lifecycle using the digital thread*. PhD thesis, Virginia Tech, 2018.

[22] T. McDermott, P. Collopy, C. Paredis, and M. Nadolski, “Enterprise system-of-systems model for digital-thread enabled acquisition, technical report serc-2018-tr-109,” tech. rep., Stevens Institute of Technology and Systems Engineering Research Center, 2018.

[23] H. von Scheel, M. von Rosing, M. Fonseca, M. Hove, and U. Foldager, *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM*, vol. 1, ch. Phase 1: Process Concept Evolution. Elsevier, 2015.

[24] T. Hedberg Jr., N. Hartman, P. Rosche, and K. Fischer, “Identified research directions for using manufacturing knowledge earlier in the product life cycle,” *Internal Journal of Production Research*, 2017.

[25] H. von Scheel, M. von Rosing, M. Fonseca, M. Hove, and U. Foldager, *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM*, vol. 1, ch. Phase 2: Process Concept Evolution. Elsevier, 2015.

[26] I. C. on Systems Engineering (INCOSE), “INCOSE systems engineering handbook: A guide for systems life cycle processes and activities, fourth edition,” 2015.

[27] T. Hedberg Jr., M. Bajaj, and J. Camelio, “Using graphs to link data across the product lifecycle for enabling smart manufacturing digital threads,” *Journal of Computing and Information Science in Engineering*, 2019.

[28] S. Mittal, M. Khan, D. Romero, and T. Wuest, “A critical review of smart manufacturing industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs),” *Journal of Manufacturing Systems*, vol. 49, pp. 194–214, 2018.

[29] D. McKinsey, “Digital manufacturing- escaping pilot purgatory,” tech. rep., McKinsey Company, 2018.

[30] A.I.A.M.C. of Excellence, “Strategic guide: Additive manufacturing data management and schema,” tech. rep., ASTM International, 2019.

[31] G. Harris and L. Caudle, “A systems approach to establishing an advanced manufacturing innovation institute,” *Systems*, 2019.

[32] P. Verhoef, T. Broekhuizen, Y. Bart, A. Bhattacharya, J. Dong, N. Fabian, and M. Haenlein, "Digital transformation: A multidisciplinary reflection and research agenda," *Journal of Business Research*, vol. 122, pp. 889–901, 2021.

[33] M. Bajaj and T. Hedberg Jr, "System lifecycle handler—spinning a digital thread for manufacturing," in *28th Annual INCOSE International Symposium*, 2018.

[34] C. Leiva, "Demystifying the digital thread and digital twin concepts," *Industry Week*, 2016.

[35] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary perspectives on complex systems*, pp. 85–113, Springer, 2017.

[36] K. Bartsch, A. Pettke, A. Hübert, J. Lakämper, and F. Lange, "On the digital twin application and the role of artificial intelligence in additive manufacturing: a systematic review," *Journal of Physics: Materials*, vol. 4, 2021.

[37] J. Osho, A. Hyre, M. Pantelidakis, A. Ledford, G. Harris, J. Liu, and K. Mykoniatis, "Four rs framework for the development of a digital twin: The implementation of representation with a FDM manufacturing machine," *Journal of Manufacturing Systems*, vol. 63, pp. 370–380, 2022.

[38] S. Frechette, "Model based enterprise for manufacturing," *National Institute of Standards and Technology (NIST)*, 2011.

[39] T. Henry and R. Steiner, "Model-based systems engineering in the real world," in *Proceedings of the 9th Model-Based Enterprise Summit (MBE 2018)*, 2018.

[40] D. Gates and M. Bremicker, "Beyond the hype: Separating ambition from reality in Industy 4.0," tech. rep., KPMG International Cooperative, 2017.

[41] K. Overstreet, "Which came first, the drawing or the building? understanding the world's first architectural processes," 2020. Retrieved April 24, 2022, from https://www. archdaily.com/.

[42] P. Booker, *A History of Engineering Drawing*. Chatto Windus, 1963.

[43] E. Andrews, “8 remarkable early maps: Explore eight of the most important maps from the early history of cartography.,” 2018. Retrieved April 24, 2022 from https://www.history.com/news/8-remarkable-early-maps.

[44] L. Hardesty, “The birth of electrical engineering: The creation of the first electrical-engineering curriculum may have said as much about MIT’s educational philosophy as it did about the pace of innovation.” Retrieved April 6, 2022, from https://news.mit.edu/2011/timeline-eecs-0309.

[45] A. Sampaio, “Historical evolution of technical drawing in engineering,” in *2018 3rd International Conference of the Portuguese Society for Engineering Education (CISPEE)*, 2018.

[46] J. Mee, “Frederick w. Taylor: American inventor and engineer.” Retrieved April 14, 2022, from https://www.britannica.com/biography/ Frederick-W-Taylor.

[47] B. Price, *BUSINESS AND ECONOMIC HISTORY, Second Series*, vol. 18, ch. Frank and Lillian Gilbreth and the Manufacture and Marketing of Motion Study, 1908-1924. Cambridge University Press, 1989.

[48] P. C. ASME Special Committee on Standardization of Therbligs and T. Symbols, *ASME Operation and Flow Process Charts*. The American Society of Mechanical Engineers, 1947.

[49] N. W. Encyclopedia, “Flowchart,” n.d. Retrieved April 24, 2022 from https://www.newworldencyclopedia.org/entry/Flowchart.

[50] B. Library, “Henry laurence gantt,” n.d. Retrieved April 24, 2022 from https://www.bl.uk/people/henry-laurence-gantt.

[51] M. Grant, “Gantt chart,” 2021. Retrieved from April 24, 2022 from https://www.investopedia.com/terms/g/gantt-chart.asp.

[52] W. Rouse and D. Bodner, “Multi-level modeling of complex socio-technical systems report no. ccse-2013-01,” tech. rep., Center for Complex Systems Enterprises, 2013.

[53] D. A. U. PRESS, “Systems engineering fundamentals,” tech. rep., Department of Defense Systems Management College, 2001.

[54] N. C., “What is control flow diagram in software engineering?,” 2022. Retrieved April 25, 2022, from https://www.mcnallyinstitute.com/ what-is-control-flow-diagram-in-software-engineering/.

[55] M. von Rosing, A. Scheer, J. Zachman, D. Jones, J. Womack, and H. von Scheel, *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM*, vol. 1, ch. Phase 3: Process Concept Evolution. Elsevier, 2015.

[56] J. Womack, “Value stream mapping: This lean tool can help companies level production, resulting in dramatic reductions in throughput time and costs, and improved quality,” *Society of Manufacturing Engineer’s Manufacturing Engineering*, 2006. Retrieved April 14, 2022, from https://www.sme.org/technologies/ value-stream-mapping/.

[57] C. Halton, “Kanban,” 2022. Retrieved April 25, 2022 from https://www. investopedia.com/terms/k/kanban.asp.

[58] L. Hennigan and C. Bottorff, “What is a kanban board? basics and benefits,” 2022. Retrieved April 25, 2022, from https://www.forbes.com/advisor/business/ software/what-is-kanban-board/.

[59] “Functional flow block diagram,” n.d. Retrieved April 25, 2022 from https:// en-academic.com/dic.nsf/enwiki/11327961.

[60] N. I. of Standards and T. (NIST), “Federal information processing standards publication: Integration definition for function modeling IDEF0),” U.S. Department of Commerce, Technology Administration.

[61] M. von Rosing, S. White, F. Cummins, and H. de Man, *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM*, vol. 1, ch. Business Process Model and Notation—BPMN. Elsevier, 2015.

[62] M. Hammer, “The process audit,” *Harvard Business Review*, 2007.

[63] B. Kulvatunyou, H. Oh, N. Ivezic, and S. Nieman, “Standards-based semantic integration of manufacturing information: Past, present, and future,” *Journal of Manufacturing Systems*, vol. 52, pp. 184–197, 2019.

[64] J. Sztipanovitz, T. Bapty, X. Koutsoukos, Z. Lattmann, S. Neema, and E. Jackson, "Model and tool integration platforms for cyber-physical system design," in *Proceedings of the IEEE Special Issue on Design Automation for CPS, 2018 IEEE*, 2018.

[65] A. Kusiak, "Service manufacturing: Basic concepts and technologies," *Journal of Manufacturing Systems*, vol. 52, pp. 198–204, 2019.

[66] M. Helu, T. Hedberg Jr., and A. Feeney, "Reference architecture to integrate heterogeneous manufacturing systems for the digital thread," *CIRP Journal of Manufacturing Science and Technology*, vol. 19, 2017.

[67] J. Mehler, S. McGee, and R. Edson, "Leveraging systemigrams for conceptual analysis of complex systems: Application to the U.S. national security system," in *8th Conference on Systems Engineering Research*, (Hoboken, NJ), 2010.

[68] T. McDermott, P. Collopy, M. Nadolski, and C. Paredis, "The future exchange of digital engineering data and models: an enterprise systems analysis," in *17th Annual Conference on Systems Engineering Research (CSER)*, 2019.

[69] R. Bonnard, J. Hascoe¨t, P. Mognol, E. Zancul, and A. Alvares, "Hierarchical object-oriented model HOOM) for additive manufacturing digital thread," *Journal of Manufacturing Systems*, vol. 50, pp. 36–52, 2019.

[70] Systems and P. E. Company, "The innoslate approach to digital engineering." Retrieved April 15, 2022 from https://specinnovations.com/wp-content/uploads/2019/05/DigitalEngineeringWhitepaper.pdf.

[71] D. Firesmith, "Mission thread analysis using end-to-end data flows - part 1," *Carnegie Mellon University Software Engineering Institute*, 2019. Retrieved April 25, 2022, from https://insights.sei.cmu.edu/blog/mission-thread-analysis-using-end-to-end-data-flows-part-1/.

[72] D. Firesmith, "Mission thread analysis using end-to-end data flows - part 1," *Carnegie Mellon University Software Engineering Institute*, 2019. Retrieved April 25, 2022 from https://insights.sei.cmu.edu/blog/mission-thread-analysis-using-end-to-end-data-flows-part-2/.

[73] L. Wheatcraft, "Everything you wanted to know about interfaces, but were afraid to ask," in *INCOSE International Symposium, July 2010*, 2010.

[74] T. Iyamu, "The architectures of data and information: Their confounded confusion," *South African Journal of Information Management*, 2019.

[75] D. Romero, P. Galardelli, D. Powell, and T. Wuest, "Digital lean cyber-physical production systems: The emergence of digital lean manufacturing and the meaning of digital waste," in *APMS - Production Management for Data-Driven, Intelligent, Collaborative, and Sustainable Manufacturing*, 2018.

[76] P. Roh, A. Kunz, and K. Wegener, "Information stream mapping: Mapping, analysing and improving the efficiency of information streams in manufacturing value streams," *CIRP Journal of Manufacturing Science and Technology*, 2019.

[77] T. Busert and A. Fay, "Information quality focused value stream mapping for the co-ordination and control of production processes," *International Journal of Production Research*, vol. 59, no. 15, pp. 4559 – 4578, 2021.

[78] D. A. Kinard, "F-35 digital thread and advanced manufacturing," in *AIAA AVIATION Forum, 2018 Aviation Technology, Integration, and Operations Conference*, 2018.

[79] P. Embry, "Digital tapestry, Lockheed Martin space systems company." Retrieved April 4, 2022 from https://sma.nasa. gov/docs/default-source/News-Documents/embry_lockheed-martin-space-systems-digital-transformation.pdf?sfvrsn=95cae0f8_4, 2016.

[80] G. Harris, "Address to the general assembly of the Model Based Enterprise Summit," (NIST Headquarters in Gaithersburg, MD), 2014.

[81] J. Gray, "Reducing new product introduction time and cost through more effective collaboration," in *NIST Technical Note 1753, Model Based Enterprise/Technical Data Pack- age Summit Report*, 2012.

[82] T. Hedberg Jr., J. Lubell, L. Fischer, L. Maggiano, and A. Feeney, "Testing the digital thread in support of model-based manufacturing and inspection," *Journal of Computing and Information Science in Engineering*, vol. 16, 2016.

[83] G. Bulter and C. Woody, "Mission threads: Bridging mission and systems engineering," 2017. Presentation retrieved online April 25, 2022 from https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2017/ systems/Wednesday/Track2/19868_Butler_Woody.pdf.

[84] M. Helu, A. Joseph, and T. Hedberg Jr., "A standards-based approach for linking as-planned to as-fabricated product data," *CIRP Ann Manuf Technol. 2018*, vol. 67, 2018.

[85] B. Abbott, T. Bapty, C. Biegl, G. Karsai, and J. Sztipanovits, "Model-based software synthesis," *IEEE Software*, vol. 10, pp. 42–52, 1993.

[86] E. Jeong, D. Jeong, and S. Ha, "Dataflow model–based software synthesis framework for parallel and distributed embedded systems," *ACM Transactions on Design Automation of Electronic Systems*, vol. 26, pp. 1–38, 2021.

[87] G. Simko, T. Levendovsky, S. Neema, E. Jackson, T. Bapty, J. Porter, and J. Sztipanovitz, "Foundation for model integration: Semantic backplane," in *Proceedings of the ASME 2012 International Design Engineering Technical Conferences Computers and Infor- mation in Engineering Conference IDETC/CIE 2012*, (Chicago, IL), 2012.

[88] J. Lubell, K. Chen, J. Horst, S. Frechette, and P. Huang, "NIST technical note 1753: Model based enterprise/technical data package summit report," tech. rep., National Institute of Standards and Technology (NIST) Engineering Laboratory and the Office of the Secretary of Defense (OSD), 2012.

[89] J. Lubell, S. Frechette, R. Lipman, F. Proctor, J. Horst, M. Carlisle, and P. Huang, "Model-based enterprise summit report," tech. rep., Army Research Laboratory (ARL), 2014.

[90] T. Hedberg Jr., "Enabling connections in the product lifecycle using the digital thread-dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of doctor of philosophy in industrial and systems engineering," 2018.

[91] D. Thomas, "NIST advanced manufacturing series 100-26, the model based enterprise a literature review of costs and benefits for discrete manufacturing," tech. rep., National Institute of Standards and Technology (NIST), 2019.

[92] M. Bowden and M. Jones, "The cost of enabling the digital thread," in *Proceedings of the 11th Model-Based Enterprise Summit (MBE 2020)*, 2020.

[93] Oracle, "What is plm (product lifecycle management)?." retrieved April 4, 2022 from https://www.oracle.com/scm/product-lifecycle-management/ what-is-plm/.

[94] Siemens, "Product lifecycle management (plm) software." retrieved April 6, 2022 from https://www.plm.automation. siemens.com/global/en/our-story/glossary/

product-lifecycle-management-plm-software/12506.

[95] PTC, "PLM for engineering: Establish a foundation for the digital thread." Retrieved April 4, 2022 from https://www.ptc.com/en/technologies/plm/ digital-thread/plm-engineering.

[96] M. Hasan and S. B., "Decentralized cloud manufacturing-as-a-service (CMAAS) platform architecture with configurable digital assets," *Journal of Manufacturing Systems*, vol. 56, pp. 157–174, 2020.

[97] S. Feng, W. Bernstein, T. Hedberg Jr., and A. Feeney, "Toward knowledge management for smart manufacturing," *Journal of Computing and Information Science in Engineering*, 2019.

[98] S. Krima, T. Hedberg, and A. Feeney, "Securing the digital threat for smart manufacturing: A reference model for blockchain-based product data traceability," *NIST Advanced Manufacturing Series 300-6*, 2019.

[99] D. Mies, W. Marsden, and S. Warde, "Overview of additive manufacturing informatics: "a digital thread"," *Integrating Materials and Manufacturing Innovation*, 2016.

[100] L. Delligatti, *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison Wesley and Pearson, 2013.

[101] N. S. R. Consortium, “Model based systems engineering (MBSE),” n.d. Retrieved April 26, 2022 from https://www.nasa.gov/consortium/ ModelBasedSystems.

[102] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SySML, The Systems Modeling Language, Third Edition*. Morgan Kaufmann and Elsevier, 2014.

[103] C. T. S. M. Institute, “Accelerating the democratization and adoption of smart manufacturing.” Retrieved July 2, 2023 from https://www.cesmii.org/.

[104] A. S. of Mechanical Engineers (ASME), “Model based enterprise (MBE),” n.d. Retrieved from https://www.asme.org/codes-standards/about-standards/ the-future-of-asme-standards/model-based-enterprise-(mbe).

[105] T. West and M. Blackburn, “Is digital thread/digital twin affordable? a systemic assessment of the cost of DoD’s latest Manhattan project,” *Procedia Computer Science*, vol. 114, pp. 47–56, 2017.

[106] J. Waters and M. Ceruti, “Modeling and simulation of information flow: A study of infodynamic quantities,” in *The 15th International Command and Control Research and Technology Symposium (ICCRTS)*, 2010.

[107] J. Waters, J. Eitelberg, R. Patel, and M. Ceruti, “Investigation of information flow in hierarchical organizations using agent-based modeling,” in *SCS/ACM Agent-Directed Simulation Symposium (ADS’09)*, 2009.

[108] T. Pond, “Measuring and modeling information flow on social networks,” 2020. A Thesis Presented by Tyson Charles Pond to The Faculty of the Graduate College of The University of Vermont.

[109] M. Evans, A. Yarbrough, G. Harris, and G. Purdy, “A simulation testbed for the evaluation of product and information flows in a manufacturing system,” in *IISE Annual Virtual Conference & Expo 2021*, 2021.

[110] D. Zhong, J. Fan, G. Yang, B. Tian, and Y. Zhang, “Knowledge management of product design: A requirements-oriented knowledge management framework based on kansei engineering and knowledge map,” *Advanced Engineering Informatics*, vol. 52, 2022.

[111] S. Jackson, “Introducing systems engineering into a traditional commercial organization,” in *INCOSE International Symposium, July 1996*, 1996.

[112] A. Yarbrough, “Data and information flow improvements in manufacturing systems- a dissertation submitted to the graduate faculty of auburn university in partial fulfillment of the requirements for the degree of doctor of philosophy,” 2022.

[113] T. West and M. Blackburn, “Is digital thread/digital twin affordable? a systemic assessment of the cost of DoD’s latest Manhattan project,” *Procedia Computer Science*, vol. 114, pp. 47–56, 2017.

[114] Techopedia, “Integration definition (IDEF)- what does integration definition (IDEF) mean?,” N.A. Article retrieved online January 31, 2023 from https://www.techopedia.com/definition/19924/ integration-definition--idef.

[115] I. Knowledge Based Systems, “IDEF- integrated definition methods (IDEF),” N.A. Website retrieved online January 31, 2023 from https://www.idef.com/.

[116] SysML., “Sysml diagram tutorial,” N.A. Website retrieved online January 31, 2023 from https://sysml.org/tutorials/sysml-diagram-tutorial/.

[117] N.A., “Uml 2.5 diagrams overview,” N.A. Website retrieved online January 31, 2023 from https://www.uml-diagrams.org/uml-25-diagrams.html.

[118] S. E. S. C. of the IEEE Computer Society, “IEEE standard for functional modeling language—syntax and semantics for IDEF0,” 1998. Standard retrieved online January 30, 2023 from https://www-ieee-org.spot.lib.auburn.edu/.

[119] M. Rother and J. Shook, *Learning to See: Value Stream Mapping to Create Value and Eliminate Muda*. Lean Enterprise Institute, 2009.

[120] P. Gill, “NASA integrated model-centric architecture,” in *NIST Technical Note 1753, Model Based Enterprise/Technical Data Package Summit Report*, 2012.

[121] “The Britannica dictionary.” Online Dictionary. Retrieved April 20, 2022 from https: //www.britannica.com/dictionary/data.

[122] J. Piatt, “5 steps to improving profitability of high-mix/low-volume production,” *Indus- try Week*, 2015.

[123] Q. Lu and B. Botha, “Process development: a theoretical framework,” *International Journal of Production Research*, vol. 44, no. 15, 2006.

[124] M. El-Sayed, “Value streaming through customer participation in product realization,” *SAE International Journal of Materials and Manufacturing*, vol. 9, no. 3, 2016.

[125] N.A., “Structured vs. unstructured data: What’s the difference?,” N.A. Online arti- cle retrieved online June 18, 2022 from https://www.ibm.com/cloud/blog/ structured-vs-unstructured-data.

[126] D. Call and D. Herber, “Applicability of the diffusion of innovation theory to accelerate model-based systems engineering adoption,” *System Engineering*, vol. 25, pp. 574–583, 2022.

[127] E. Carroll and R. Malins, “Sandia report- sand2016-2607: Systematic literature review: How is model-based systems engineering justified?,” tech. rep., Sandia National Laboratories, 2016.

[128] K. Henderson and A. Salado, “Value and benefits of model-based systems engineering (MBSE): Evidence from the literature,” *System Engineering*, vol. 24, pp. 51–66, 2020.

[129] K. Campo, T. Teper, C. Eaton, A. Shipman, G. Bhatia, and B. Mesmer, “Model-based systems engineering: Evaluating perceived value, metrics, and evidence through literature,” *System Engineering*, vol. 26, pp. 104–129, 2022.

[130] U. S. of America Department of Defense, “Mil-std 3022 w/change 1: Department of defense standard practice- documentation of verification, validation, and accreditation (VV&A) for models and simulations,” tech. rep., United States of America Department of Defense, 2012.

[131] U. S. of America Department of Defense, “Model & simulation verification, validation, and accreditation recommended practices guide,” tech. rep., United States of America Department of Defense, N.A.

[132] M. Blackburn, D. Verma, R. Dillon-Merrill, R. Blake, M. Bone, B. Chell, R. Dove, J. Dzielski, P. Grogan, S. Hoffenson, E. Hole, R. D. Jones, B. Kruse, K. Pochiraju, C. Synder, I. Cloutier, R. Grosse, and T. Hagedorn, "Transforming systems engineering through model-centric engineering: A013 interim technical report serc-2017-tr-111," tech. rep., Systems Engineering Research Center, 2018.

[133] J. Kasser, "A framework for understanding systems engineering," *INCOSE*, 2007.

[134] D. Rousseau, "A framework for understanding system principles and methods," in *28th Annual INCOSE International Symposium*, 2018.

[135] D. Sturdevant, "(still) learning from Toyota," 2014. Online article retrieved online September 05, 2022 from https://www.mckinsey. com/industries/automotive-and-assembly/our-insights/still-learning-from-toyota.

[136] M. Hargave, "Kaizen: Understanding the Japanese business philosophy," 2022. Retrieved April 22, 2023 from https://www.investopedia.com/terms/k/ kaizen.asp.

[137] S.E.S.C. of the IEEE Computer Society, "IEEE std 1233, 1998 edition, IEEE guide for developing system requirements specifications," tech. rep., IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board, 1998.

[138] S.E.S.C. of the IEEE Computer Society, "IEEE std 1362, 1998 edition, IEEE guide for information technology system definition-concept of operations (CONOPS) document," tech. rep., IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board, 1998.

[139] T. Ohno, *Toyota production system: Beyond large-scale production.* Productivity Press, 1988.

[140] T. Blog, "Muda, muri, mura - Toyota production system guide," 2013. Retrieved September 06, 2022 from https://mag.toyota.co.uk/ muda-muri-mura-toyota-production-system/.

[141] W. Kenton, "Hawthorne effect definition: How it works and is it real," 2022. Retrieved March 26, 2023 from https://www.investopedia.com/terms/h/ hawthorne-effect.asp.

[142] M. Ceruti and S. Rubin, “Infodynamics: Analogical analysis of states of matter and information,” *Information Sciences*, vol. 117, pp. 969–987, 2007.

[143] D. Ferguson, “Therbligs: The keys to simplifying work.” Retrieved July 8, 2023 from https://web.mit.edu/allanmc/www/Therblgs.pdf.

Appendix

2. Q/A for Activity:

- Qa. What activity are you involved in directly before the previous activity in Functional Area X (If multiple, pick one and then do the other/s in another response section)?
    i. A. .....

- Qb. Please offer a brief description of the activity (high-level summary):
    i. A. ......

- Qc. Under which conditions does this activity occur (i.e., when is this activity necessary)?
    i. A. ....

- Qd. What V&V decisions are made in this activity, and what data vessels are associated with these decisions?
    i. A. ....

- Qe. Please provide estimates for the minimum, maximum, and most likely timeline of the activity.
    i. Minimum: ......
    ii. Maximum: ......
    iii. Most Likely: ......

- Qf. What activity/s directly precede this activity?
    i. A. ......

- *Qg. What activity/s directly succeed this activity?*
    i. *A. ....*

- *Qh. What Data Vessels were inputted into this activity?*

Initial DV Inputs

i. *Data Vessel Input 1:*
    - *Data Vessel Name:*
    - Personnel who acquired data for activity:
    - Place data was retrieved from:
    - Format data was captured in:

ii. *Data Vessel Input X:*
    - Data Vessel Name:
    - Personnel who acquired data for activity:
    - Place data was retrieved from:
    - Format data was captured in:

Figure A1: Second Application Standardized Interview Template Part 1

Iterative &/or Conditional DV Inputs

iii. *Data Vessel Input X:*
- Data Vessel Name:
- Personnel who acquired data for activity:
- Place data was retrieved from:
- Format data was captured in:

iv. *Data Vessel Input X:*
- Data Vessel Name:
- Personnel who acquired data for activity:
- Place data was retrieved from:
- Format data was captured in:

➢ *Qi. What Data Vessels were outputted from this activity?*

Iterative &/or Conditional DV Outputs

i. *Data Vessel Output 1:*
- Data Vessel Name:
- Personnel who created and/or handled data:
- Tools used to manipulate data:
- Format data was captured in:
- Place data was stored:
- Personnel the data was sent to:
- Method in which data sent to personnel:

ii. *Data Vessel Output X:*
- Data Vessel Name:
- Personnel who created and/or handled data:
- Tools used to manipulate data:
- Format data was captured in:
- Place data was stored:
- Personnel the data was sent to:
- Method in which data sent to personnel:

Final DV Outputs

iii. *Data Vessel Output X:*
- Data Vessel Name:
- Personnel who created and/or handled data:
- Tools used to manipulate data:
- Format data was captured in:
- Place data was stored:
- Personnel the data was sent to:
- Method in which data sent to personnel:

Figure A2: Second Application Standardized Interview Template Part 2

Q/A for Example Activity 4.3:

- Qa. What activity are you involved in directly before the previous activity in Functional Area 4 (If multiple, pick one and then do the other/s in another response section)?
    i. A. Activity 4.3 Conduct Requirements Review

- Qb. Please offer a brief description of the activity (high-level summary):
    i. A. Design requirements are reviewed for approval by Engineer 1, Engineer 2, and Management 1.

- Qc. Under which conditions does this activity occur (i.e., when is this activity necessary)?
    i. A. Always happens in this system after Activity 3.3.

- Qd. What V&V decisions are made in this activity, and what data vessels are associated with these decisions?
    i. A. The requirements verification decision is made by reviewing the Requirements Verification Form.

- Qe. Please provide estimates for the minimum, maximum, and most likely timeline of the activity.
    i. Minimum: 2 hours
    ii. Maximum: 5 days
    iii. Most Likely: 3 days

- Qf. What activity/s directly precede this activity?
    i. A. Activity 3.3 Conduct Requirements Verification and 4.2 Meeting X

- *Qg. What activity/s directly succeed this activity?*
    i. *A. Activity 3.4 Conduct Requirements Validation*

- *Qh. What Data Vessels were inputted into this activity?*

Initial DV Inputs

i. *Data Vessel Input 1:*
    - Data Vessel Name: Meeting Invitation to set up initial Requirements Review meeting and to send Initial Requirements Draft.
    - Personnel who acquired data for activity: Engineer 1
    - Place data was retrieved from: Email
    - Format data was captured in: Email text

ii. *Data Vessel Input 2:*
    - Data Vessel Name: Requirements Review Form Template
    - Personnel who acquired data for activity: Engineer 1
    - Place data was retrieved from: Organizational Portal
    - Format data was captured in: Word document

iii. *Data Vessel Input 3:*
    - Data Vessel Name: Initial Requirements Draft
    - Personnel who acquired data for activity: Engineer 1
    - Place data was retrieved from: Engineer's Computer Drive
    - Format data was captured in: Word document

Figure A3: Generic Example of Second Application Completed Questionnaire Part 1

Iterative &/or Conditional DV Inputs

iv. *Data Vessel Input 4:*
- Data Vessel Name: Meeting Invitation to set up next Requirements Review meeting and to send Requirements Draft (if updated). **[if Requirements Review not complete yet] [iterative].**
- Personnel who acquired data for activity: Engineer 1, Engineer 2, Management Rep. 1
- Place data was retrieved from: Email
- Format data was captured in: Email Text

v. *Data Vessel Input 5:*
- Data Vessel Name: Draft Requirements Review Form **[if Requirements Review not complete yet] [iterative].**
- Personnel who acquired data for activity: Engineer 1
- Place data was retrieved from: Engineer's Computer Drive
- Format data was captured in: Word document

vi. *Data Vessel Input 6:*
- Data Vessel Name: Initial Requirements Draft (potentially updated) **[if Requirements Review not complete yet] [iterative].**
- Personnel who acquired data for activity: Engineer 1, Engineer 2, and Management Rep.
- Place data was retrieved from: Engineer 1, Engineer 2, and Management Rep. 1's Computer Drives
- Format data was captured in: Word document

vii. *Data Vessel Input 7:*
- Data Vessel Name: Personal Notes of conversation from Requirements Review **[if Requirements Review not complete yet] [iterative].**
- Personnel who acquired data for activity: Engineer 1, Engineer 2, and Management Rep. 1
- Place data was retrieved from: Engineer 1, Engineer 2, and Management Rep. 1's Personal Notes
- Format data was captured in: Word document or Pen and Paper

➢ *Qi. What Data Vessels were outputted from this activity?*

Iterative &/or Conditional DV Outputs

i. *Data Vessel Output 1:*
- Data Vessel Name: Meeting Invitation to set up next Requirements Review meeting and to send Initial Requirements Draft (if updated). **[if Requirements Review not complete yet] [iterative].**
- Personnel who created and/or handled data: Engineer 1
- Tools used to manipulate data: Email
- Format data was captured in: Email text
- Place data was stored: Email
- Personnel the data was sent to: Engineer 2 and Management Rep. 1
- Method in which data sent to personnel: Email

ii. *Data Vessel Output 2:*
- Data Vessel Name: Draft Requirements Review Form **[if Requirements Review not complete yet] [iterative].**

Figure A4: Generic Example of Second Application Completed Questionnaire Part 2

- Personnel who created and/or handled data: Engineer 1
- Tools used to manipulate data: Word
- Format data was captured in: Word document
- Place data was stored: Engineer 1's Computer Drive
- Personnel the data was sent to: N/A
- Method in which data sent to personnel: N/A

iii. *Data Vessel Output 3:*

- Data Vessel Name: Initial Requirements Draft (potentially updated) **[if Requirements Review not complete yet] [iterative].**
- Personnel who created and/or handled data: Engineer 1, Engineer 2, and Management Rep.
- Tools used to manipulate data: Word
- Format data was captured in: Word Document
- Place data was stored: Engineer 1, Engineer 2, and Management Rep. 1's Computer Drives
- Personnel the data was sent to: Engineer 2 and Management Rep. 1
- Method in which data sent to personnel: Email

iv. *Data Vessel Output 4:*

- Data Vessel Name: Conversation from Requirements Review **[if Requirements Review not complete yet] [iterative].**
- Personnel who created and/or handled data: Engineer 1, Engineer 2, Management Rep. 1
- Tools used to manipulate data: N/A
- Format data was captured in: Auditory
- Place data was stored: Hearer's Memory
- Personnel the data was sent to: N/A
- Method in which data sent to personnel: N/A

v. *Data Vessel Output 5:*

- Data Vessel Name: Personal Notes of conversation from Requirements Review **[if Requirements Review not complete yet] [iterative].**
- Personnel who created and/or handled data: Engineer 1, Engineer 2, Management Rep. 1
- Tools used to manipulate data: Word or Paper and Pen
- Format data was captured in: Word document or Pen and Paper
- Place data was stored: Engineer's, Engineer 2, and Management Rep. 1's Personal Notes
- Personnel the data was sent to: N/A
- Method in which data sent to personnel: N/A

<u>Final DV Outputs</u>

vi. *Data Vessel Output 6:*

- Data Vessel Name: Meeting Invitation to set up next Requirements Review meeting and to send Initial Requirements Draft (if updated). **[once Requirements Review complete]**
- Personnel who created and/or handled data: Engineer 1, Engineer 2, Management Rep. 1
- Tools used to manipulate data: Email
- Format data was captured in: Email text
- Place data was stored: Email
- Personnel the data was sent to: N/A
- Method in which data sent to personnel: N/A

Figure 6.A5: Generic Example of Second Application Completed Questionnaire Part 3

vii. *Data Vessel Output 7:*

- Data Vessel Name: Completed Requirements Review Form **[once Requirements Review complete]**
- Personnel who created and/or handled data: Engineer 1
- Tools used to manipulate data: Word
- Format data was captured in: Word document
- Place data was stored: Engineer 1's Computer Drive
- Personnel the data was sent to: N/A
- Method in which data sent to personnel: N/A

viii. *Data Vessel Output 8:*

- Data Vessel Name: Initial Requirements (potentially updated) **[once Requirements Review complete]**
- Personnel who created and/or handled data: Engineer 1, Engineer 2, and Management Rep.
- Tools used to manipulate data: Word
- Format data was captured in: Word Document
- Place data was stored: Engineer 1, Engineer 2, and Management Rep. 1's Computer Drives
- Personnel the data was sent to: Engineer 2 and Management Rep. 1
- Method in which data sent to personnel: Email

ix. *Data Vessel Output 9:*

- Data Vessel Name: Conversation from Requirements Review **[once Requirements Review complete]**
- Personnel who created and/or handled data: Engineer 1, Engineer 2, Management Rep. 1
- Tools used to manipulate data: N/A
- Format data was captured in: Auditory
- Place data was stored: Hearer's Memory
- Personnel the data was sent to: N/A
- Method in which data sent to personnel: N/A

x. *Data Vessel Output 10:*

- Data Vessel Name: Personal Notes of conversation from Requirements Review **[once Requirements Review complete]**
- Personnel who created and/or handled data: Engineer 1, Engineer 2, Management Rep. 1
- Tools used to manipulate data: Word or Paper and Pen
- Format data was captured in: Word document or Pen and Paper
- Place data was stored: Engineer's, Engineer 2, and Management Rep. 1's Personal Notes
- Personnel the data was sent to: N/A
- Method in which data sent to personnel: N/A

xi. *Data Vessel Output 11:*

- Data Vessel Name: Meeting Invitation to set up initial Requirements Review meeting and to send Initial Requirements Draft.
- Personnel who created and/or handled data: Engineer 1, Engineer 2, Management Rep. 1
- Tools used to manipulate data: Email
- Format data was captured in: Email text
- Place data was stored: Email
- Personnel the data was sent to: N/A
- Method in which data sent to personnel: N/A

Figure A6: Generic Example of Second Application Completed Questionnaire Part 4

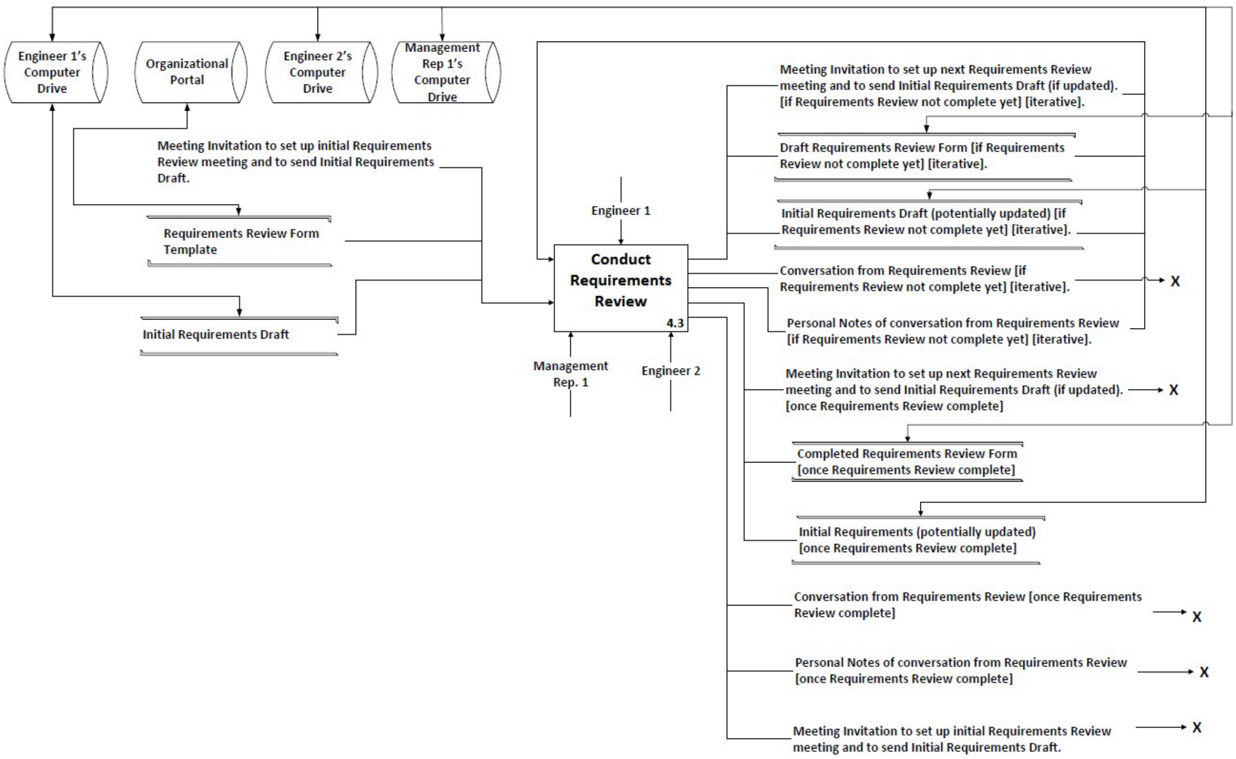


Figure A7: Data Vessel Mapping of Generic Example of Second Application Completed Questionnaire

<u>Pseudocode for the class definition file of the code:</u>

Class DataElementInstance:

def _init_ (self, data from row in excel sheet):

initialize instance id, data element id, data element name, functional activity,

data vessel name, relationships, vessel type, storage location, actors 1,2, and 3,

manual transfer, new data element, labor hours, form waste flag, visited flag,

excess_1 visit flag, excess_2 visit flag.

set unstructured data forms to contain recognized types

set digital forms to contain recognized types

set parents, children to empty dictionaries

set base parents to empty list

call method determine_base_parents()

call method check_form_waste()

```
def check_form_waste():
    if node.data_format in unstructured data forms:
        set form_waste_flag to true


def determine_base_parents():
    look at the relationships in the row data associated to the node.
    break up all the elements to determine who the parents are to the node.
        while preserving the priority in which each parent should be seen
    add a dictionary to each parent defining the type of relationship ("and" or "or") with
        each waste flag set to false


def determine_waste(counter_list)
    if node has not been visited yet:
        set visited flag to True
        for all children of the current node:
            perform waste check for storage location
                if there is waste set storage waste flag
            perform waste check for change waste
                if there is waste set change waste flag
            perform waste check for manual waste
                if there is waste set manual waste flag
            perform waste check for digital to paper
                if there is waste set digital to paper waste flag
            perform waste check for auditory to paper
                if there is waste set auditory to paper waste flag
            perform waste check for actor access dv
                if there is waste set actor access dv  waste flag
            perform waste check for actor access de
                if there is waste set actor access de waste flag
            perform waste check for excess 1 waste
```

```
                    if there is waste set excess 1  waste flag
                perform waste check for excess 2 waste
                    if there is waste set excess 2 waste flag
                recursive call to determine_waste(current child of node)
        return counter_list


    def add_parent(parent id, parent object)
        add parent to current node.parents with reference to its memory location


    def add_child(self, child_id, child_object):
         add child to current node.children with reference to its memory location


    def update_linked_relationships():
        create relationship in of all parents to the current node as a string
```

<u>Pseudocode for the main file of the code:</u>

```
create all dictionaries and lists that will be used to contain information about nodes and wastes
for every row in the excel file:
    if the row is the first row of excel sheet:
        get column names
    else:
        create node and store it in dictionary

        if the node's data element is not in unique forms waste:
            create a spot in unique forms waste for it equal to 1
        else:
            increment that instance by 1
```

```
        if the node's storage location is not identified as unique separation:
            create a spot in unique separation waste for it equal to 1
        else:
            increment that instance by 1

        if the node's data element name is not identified in level of automation for data elements:
            create a dictionary spot for that data element
            if it is a manual transfer:
                set that data element instance in level of automation equal to 1
        elif it is a manual transfer:
            increment that data element instance in level of automation equal by 1

        if node's data element name not accounted for in data element instances:
            create a spot in data element instances equal to 1
        else:
            increment that data element name in data element instances by 1

        if node's data element name not accounted for in labor hours:
            create a spot in labor hours equal to 1
        else:
            add the associated labor hours to the spot corresponding to the data element name

        if the node storage location is not in the centrality index dictionary:
            give it a corresponding location equal to 1
        else:
            increment that corresponding instance by 1

        store memory location of node in node dictionary
        add new instance to node lists
    increment index by 1
```

```
for all data element instances:
    if the data element instance's storage is <= the storage baseline
        add corresponding data element instance as a storage waste

for every node:
    for every parent of the node:
        get the grandparent(in relation to current node)
        add memory location between parent and current node
        increment relationship counter by 1
        add memory location of current node to the parent node
    if there are no parents for current node:
        mark that current node is a root node

total number of relationships = relationship counter
create initial list to hold the output
for every root node:
    counter list = current node.determine_waste(list)

perform calculations with values from counter list

for data element instance in excess waste:
    excess total += excess waste[data element instance]

perform calculations for metrics
```