Fall 11-2020

Model Based Systems Engineering for a Venture Class Launch Facility

Walter McGee Taraila
Old Dominion University, wtaraila@gmail.com

Recommended Citation
Taraila, Walter M.. "Model Based Systems Engineering for a Venture Class Launch Facility" (2020). Master of Science (MS), Thesis, Mechanical & Aerospace Engineering, Old Dominion University, DOI: 10.25777/b713-zf77
https://digitalcommons.odu.edu/mae_etds/326

MODEL BASED SYSTEMS ENGINEERING FOR A VENTURE CLASS LAUNCH FACILITY

by

Walter McGee Taraila
B.Sc. May 2012, University of Maryland, College Park

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

MECHANICAL AND AEROSPACE ENGINEERING

OLD DOMINION UNIVERSITY
December 2020

Approved by:

Sharan Asundi (Director)

Holly Handley (Member)

Miltos Kotinis (Member)

ABSTRACT

MODEL BASED SYSTEMS ENGINEERING FOR A VENTURE CLASS LAUNCH FACILITY

Walter McGee Taraila
Old Dominion University, 2020
Director: Dr. Sharan Asundi

A study of Model-Based Systems Engineering (MBSE) applied to a small-lift launch facility is presented. The research uses Systems Modeling Language (SysML) products and functional diagrams to document the structure, controls, electrical power, hydraulic, safety mechanisms, software, and fluid ground systems on a launch pad. The research is motivated by the need to design complex systems with an unambiguous understanding that improves communication, quality, productivity, and reduces risk. A model is developed following the ISO/IEC-15288 technical process framework. The stakeholder requirements are defined and analyzed to provide traceability to individual systems and subsystems. An architectural design is realized and implemented by generating engineering artifacts such as Piping and Instrumentation drawings (P&ID) and a hydraulic circuit diagram. The architecture is verified and validated by performing engineering trade studies focused on the fuel and pneumatic systems.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor Dr. Sharan Asundi for his help and collaboration while developing this paper. I would also like to extend my gratitude to the Mid-Atlantic Regional Spaceport for giving me the opportunity to work on launch pads. The first-hand experience working on aerospace systems with knowledgeable engineers is truly invaluable. I am grateful for Thomas Reiker taking time out of his busy schedule to teach me more about hydraulic control systems. Special thanks to Zach Quandt and Jay Richardson for their guidance, patience, and encouragement while I continue to learn multiple engineering disciplines. I would like to acknowledge Heidi Glaudel for her help while learning how to use SysML. I very much appreciate Dr. Miltos Kotinis and Dr. Holly Handley for their willingness to review my thesis and participate in the defense. I would like to thank Alina Vartan for being an instrumental reviewer and editor who was always willing to provide input. This achievement would not be possible without the motivation, inspiration, and constant support of my friends and family. I am especially thankful to my mother, Karen, and father, Walt, who gave me all of the love and reinforcement I needed to accomplish this goal. I could not have done this without all of you.

NOMENCLATURE

ACT	Activity Diagram
API	Application Programming Interface
BDD	Block Definition Diagram
CAD	Computer-Aided Design
EGSE	Electrical Ground Support Equipment
ET	External Tank
FGSE	Fluid Ground Support Equipment
GHe	Gaseous Helium
GN_2	Gaseous Nitrogen
GO_2	Gaseous Oxygen (GOX)
HVAC	Heating, Ventilation, and Air Conditioning
IBD	Internal Block Diagram
INCOSE	International Council on Systems Engineering
LEO	Low Earth Orbit
LN_2	Liquid Nitrogen
LOTO	Lock-Out Tag-Out
LO_2	Liquid Oxygen (LOX)
LSP	Launch Service Provider
LV	Launch Vehicle
MARS	Mid-Atlantic Regional Spaceport
MAWP	Maximum Allowable Working Pressure
MBSE	Model-Based Systems Engineering

MEOP	Maximum Expected Operating Pressure
MoE	Measure of Effectiveness
MoP	Measure of Performance
MoS	Measure of Suitability
MSDS	Material Safety Data Sheet
NEC	National Electric Code
O&M	Operations and Maintenance
OO	Object-Oriented
OOSEM	Object-Oriented Systems Engineering Method
OSHA	Occupational Safety and Health Administration
PRV	Pressure Relief Valve
PT	Pressure Transducer
SD	Sequence Diagram
SME	Subject Matter Expert
SSO	Sun Synchronous Orbit
ST	Storage Tank
STM	State Machine Diagram
SysML	Systems Modeling Language
TPM	Technical Performance Measures
UC	Use Case Diagram
UML	Unified Modeling Language
VTVL	Vertical Take-Off Vertical Landing

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

1. INTRODUCTION

1.1 Inspiration

NASA defines systems engineering as “a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system” [33]. Model-based systems engineering (MBSE) is an approach to systems engineering where the process is driven by a model representing the system. The design, analysis, specifications, and verification information are captured within the MBSE model in order to describe the system. The model of the system is maintained and controlled throughout the project’s life cycle in order to offer system engineers a consistent and precise characterization of the system which is traceable to the requirements of the mission at hand.

The prevalent approach for modern-day systems engineering and architecting entails the creation of documentation in the form of a disjointed set of texts, diagrams, spreadsheets, etc., all of which are managed to keep site leads and fellow engineers on the project abreast of changes in the system. As the project’s design evolves, it is the responsibility of subject matter experts to maintain all of these artifacts in an “As-Built” form in order for technicians, fellow engineers, and users to properly preserve and operate the systems. The MBSE approach proposes a replacement of the document-based method by creating a single system model which integrates all of the information that was formerly documented in a disjointed set of documentation. MBSE is enabled by using graphical modeling languages, with Systems Modeling Language (SysML) (OMG, 2015a) as the de-facto standard. Within a SysML model, systems are abstracted offering overviews of components, interfaces, constraints, and interconnections between other systems.

The utilization of a system model can offer significant benefits over the document-based historical approach.

In the document-based systems engineering method, difficulties arise when system engineers must maintain consistency, traceability, and precision while referencing and supporting information distributed across countless artifacts. Advantages of the MBSE method compared to the document-based method include: [18]

1. System specification and design precision improvement with fewer propagating errors downstream;
2. System requirement, analysis, design, and verification traceability enhancements to improve the integrity of the system design;
3. Ability to maintain and evolve design baseline and system specifications in an improved manner throughout the project's life cycle;
4. Ability to reuse models across multiple similar projects;
5. Enhanced understanding of the entire system by each stakeholder in order to reduce miscommunication among the team.

Traditionally, systems engineering has included the use of and reference to many kinds of models. The current emphasis for MBSE targets to accomplish an overarching goal of building an integrated system model that provides multiple views of the system and subsystems. Engineering design is improved when there is consistency between design elements. Consider the state of engineering drawings prior to the emergence of parametric computer-aided design (CAD) as an analogy. Prior to CAD, engineering designs were historically documented in hand-written forms or within isolated files on a computer. If one dimension of a part was modified on a single engineering drawing, this cascaded throughout the entire project design. The change drove each individual file to be reviewed and updated in order to maintain consistency. With the invention of parametric CAD, hardware could now be linked with specified geometric relationships. One change to the system autonomously propagates throughout the entire model and alerts the designer if there is an obstruction or interference. The adoption of MBSE methodologies is projected to offer similar improvements to systems engineering [27]. Figure 1 generated by Microsoft shows that within the last century, the amount of time it has taken for significant inventions to progress from a prototype to having significant worldwide use has dramatically decreased. Cost, speed, and quality optimization requires revolutionary changes in development methods, and this is an inspiration for detailing future engineering projects using an MBSE approach. In a world of increasing efficiency, the early life cycle stages (concept, development, production) contract in order to reach the later phases (production, utilization).

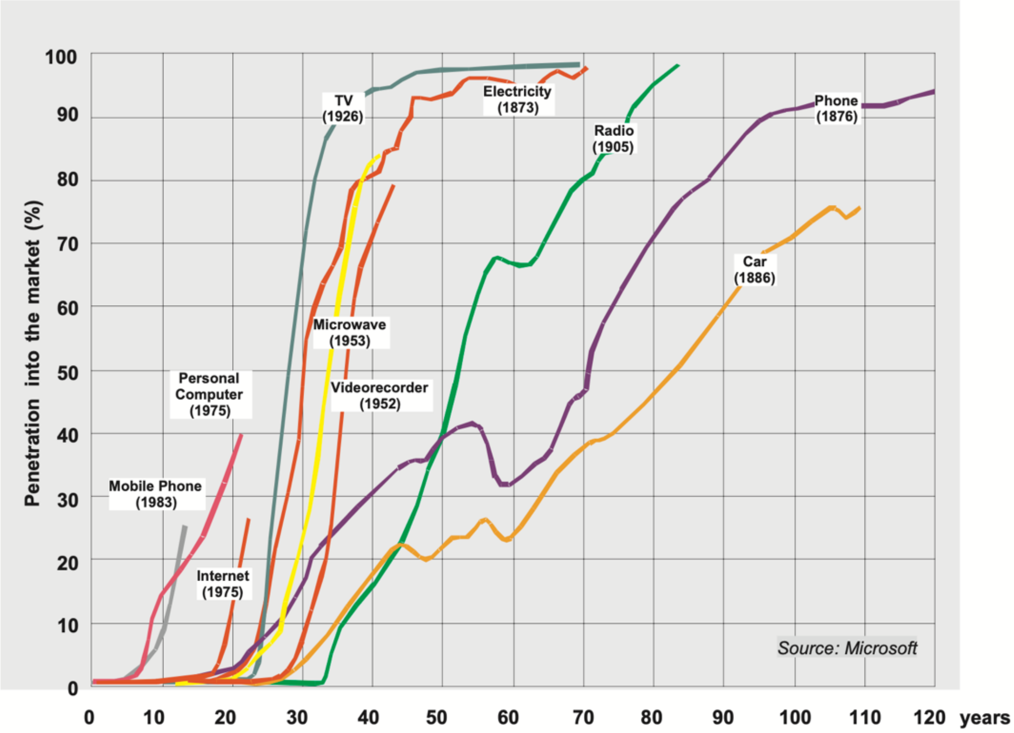


Figure 1: Time from Prototype to Significant Market Penetration [Microsoft]

1.2 Launch Pad Terminology

A launch pad is an example of a systems engineering project in which the facility's primary mission is to provide all of the resources required to launch a rocket vertically from the ground. The facility includes a launch mount to physically support the rocket and service structures for umbilical mates and all of the infrastructure that the launch vehicle requires prior to liftoff. These systems typically include civil mechanical systems, controls, fluid ground support, cryogenics, propellants, deluge, pneumatics, environmental controls, safety controls, electrical power systems, and hydraulics. The design and maintenance of these different systems brings together a diverse mixture of engineering disciplines. The systems must function cohesively in parallel for the overarching goal of getting the payload into orbit. Within each of the systems, there are subsystems that further define the launch pad.

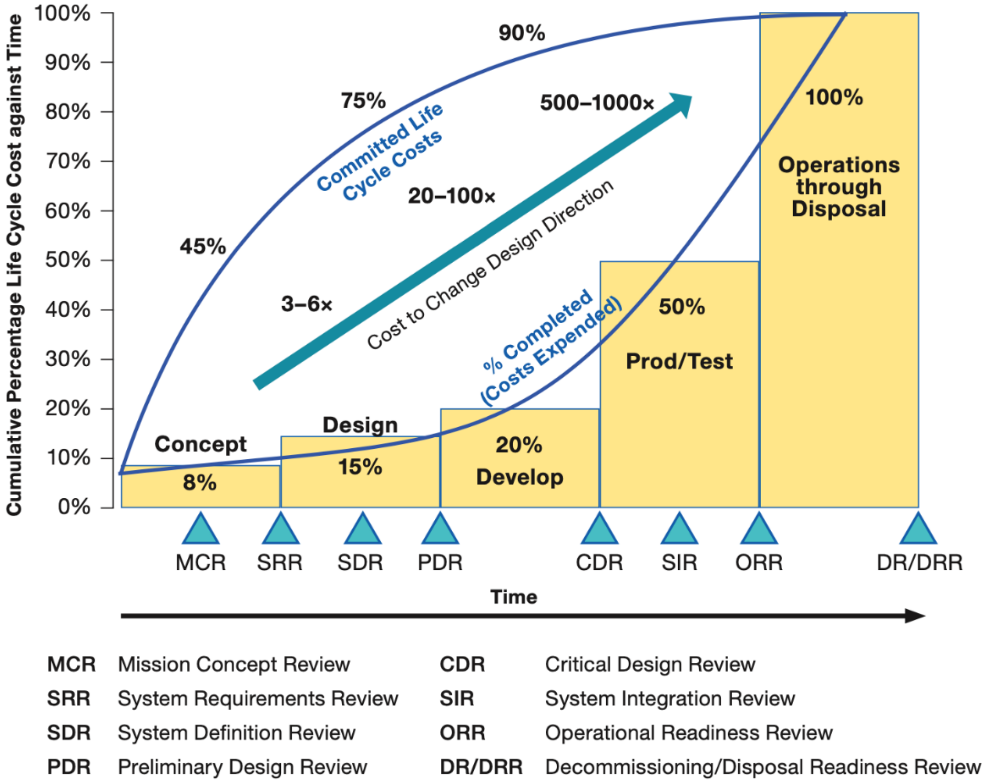
Mechanical and civil engineers primarily focus on the facility's infrastructure, the launch mount, buildings and architectural systems, which are designed to handle the full load of a rocket launch. The forces and moments which will be applied to the pad are carefully inspected during the preliminary design and linked specifically to a known rocket with a planned cadence or launch frequency. Controls and software engineers design the network, tying together thousands of individual signals from all of the other disciplines and relay these inputs and outputs (I/O) to Human Machine Interfaces (HMIs) that provide the ability to safely fuel and monitor the rocket and its surrounding infrastructure on the ground. Fluid Ground Support engineers are familiar with the characteristics and dynamics of all the commodities that interact with and support the launch vehicle such as cryogenics, propellants, water deluge systems, high pressure gases, and environmental control systems. Safety and environmental engineers are concerned with monitoring all of the hazards that may be present, and must be well aware of all designs, procedures, and systems in place to ensure that there are no injuries to personnel or hardware. Electrical engineers design and sustain the main distribution power systems, lightning protection systems, backup power systems, and interface with other groups such as FGSE to assist when high powered applications are required such as pumps, VFDs, and motors. Hydraulics are used to lift and lower the rocket to the horizontal and vertical positions, motivating these engineers to have solid electrical and mechanical backgrounds. This is by no means an exhaustive list, but it gives an indication of the complexity that goes into a launch facility. Systems engineering is an integrative and holistic discipline. The contributions of these different engineers are evaluated and balanced to create a logical whole that is not dominated by a single perspective.

The model-based approach is pervasive in all of these engineering disciplines which include mechanical, electrical, software, and control design. In a document-based approach, all subject matter experts must have meetings frequently in order to have a shared and up-to-date vision and understanding of the combined system. The system model's intent is to integrate with other models used by system engineers across all disciplines. An integrated framework for models spanning multiple disciplines is accomplished by utilizing the combined MBSE model.

1.3 Research Objectives

The main objective of systems engineering projects is to design, construct, and operate safely while accomplishing mission objectives in the most efficient way possible by considering performance, cost, risk, and schedule. Early in the design and development, life cycle costs of a project tend to get "locked in". Late identification and repair to problems cost considerably more than problems caught early in the life cycle. Descopes to mission requirements accepted later versus earlier in the project life cycle also result in reduced cost savings. Figure 2, obtained from the NASA Systems Engineering Handbook and generated by the Defense Acquisition

University, represents how costs are committed throughout the stages of a project's design starting with concept reviews and ending with decommissioning.



Adapted from INCOSE-TP-2003-002-04, 2015

Figure 2: Life Cycle Cost Impacts from Early Phase Decision-Making

When a problem is first defined, the largest degrees of freedom and greatest number of possible solutions exist. As design decisions are made, the number of potential solutions to the problem decreases and life cycle costs are solidified. Figure 2, displaying life cycle cost impacts from early phase decision-making, enforces the idea that the early stages of concept design are some of the most important times within a project's development. The exact numbers associated with cost will vary from project to project, but the general shape of the graph and the message portrayed will be similar. The systems engineer should make critical project-related information available to all key decision-makers as early in the life cycle as possible to help ensure the most cost-effective options are implemented. Efficient systems engineering processes that define highly complex systems, such as MBSE methods, improve precision, traceability, maintainability, reuse, and offer a shared understanding. While there are many references for model-based system engineering, there were no examples tailored to a rocket launch facility found during the literature review. A primary objective of this thesis is to investigate and build an MBSE framework for a venture class launch pad using the SysML language. The four pillars of SysML are requirements, structure, behavior, and parametric relationships. Each of these four pillars will be examined and applied towards the systems on a liquid-based launch pad. There are nine types of SysML diagrams that may be used to present aspects of a launch pad system. The diagrams include the Requirement Diagram, four types of Behavior Diagrams, two types of Structure Diagrams, the Parametric Diagram, and finally the Package Diagram. A launch pad is an example of a highly complex systems engineering project where all nine types of SysML diagrams are applicable and offer improvements over their document-based alternatives. The generic models explored herein may be tailored to numerous other launch sites to improve the efficiency and maintainability of similar projects. Using SysML, the following aspects of a launch pad may be described [18]:

1. A hierarchy of systems, subsystems, and components creating a system breakdown;
2. System, subsystem, and component interconnections;
3. System and component behaviors with respect to the actions these elements perform, as well as their inputs, outputs, and control flows;
4. A sequence of message exchanges between different elements to represent behavior;

5. State and state transition to further represent a system and elements or components;
6. Attributes or properties of the system, subsystem, and components, with parametric relationships to bind these elements together;
7. Requirements in a text-based format to represent the mission, and the traceability of these relationships to additional requirements, analyses, designs, and verification methods.

1.4 Research Approach and Strategy

When tasked with developing a large and multifaceted system of systems, it is imperative that the project follows the best practices of systems engineering. These guidelines and methodologies are documented and maintained by organizations such as the International Organization for Standardization (ISO) and INCOSE. MBSE methods developed by these organizations drive the specification, design, analysis, and verification of complex systems. ISO/IEC-15288 is the world-wide standard on System Life Cycle Processes [20], published by ISO. The life cycle stages are concept, development, production, utilization, support, and retirement. The standard defines a framework of technical processes outlined in Table 1 to define requirements, develop an architectural design and realize a solution that meets the requirements.

Table 1: ISO/IEC 15288 Technical Processes [49]

Index	ISO/IEC 15288 - Technical Process
A)	Stakeholder Requirements Definition
B)	Requirements Analysis
C)	Architectural Design
D)	Implementation
E)	Integration
F)	Verification
G)	Transition
H)	Validation
I)	Operation and Maintenance
J)	Disposal

ISO/IEC-15288 is an international standard organized in five groups which are Agreement, Enterprise, Project, Technical, and Special. Table 1 outlines the technical group of systems engineering processes. The heritage of this standard evolved from previous engineering efforts looking to standardize systems engineering processes, including systems engineering management (IEEE-1220) (IEEE 1998) and systems development (EIA-632) (EIA 1999). The Object-Oriented Systems Engineering Method (OOSEM), developed in 1998, is often implemented in conjunction with SysML to provide a best practice to MBSE. Present-day model-based methodologies are dominated by the object-oriented approach. The term ‘object-oriented’ (OO) is derived from the third generation of software programming languages, succeeding assembly and machine-code. A higher level of abstraction is obtained in OO with the introduction of classes, objects, aggregation, and inheritance. OOSEM outlines a framework founded on model-based and object-oriented techniques while following traditional system engineering practices. Similar to ISO/IEC-15288, technical processes are defined in OOSEM and represented in Table 2. The processes are iterative and recursive in nature, matching the intentions of the ISO/IEC-15288 technical processes [39].

Table 2: OOSEM Technical Processes [39]

Index	OOSEM - Technical Process
A)	Analyze Needs
B)	Define System Requirements
C)	Define Logical Architecture
D)	Synthesize Allocated Architectures
E)	Optimize Architecture
F)	Evaluated Alternatives
G)	Verification of Systems
H)	Validation of Systems

An executive level outline of the OOSEM method is shown by utilizing an activity diagram in Figure 3. ISO/IEC-15288 is a guide to determine what needs to be done, while OOSEM defines how that can be done, and at their intersection lies the design process outlined in this paper. While the steps are in sequence, the functions are expected to be executed in parallel and iterated multiple times prior to design completion. After multiple iterations of this outline, model artifacts are produced and combined to constitute an MBSE model.

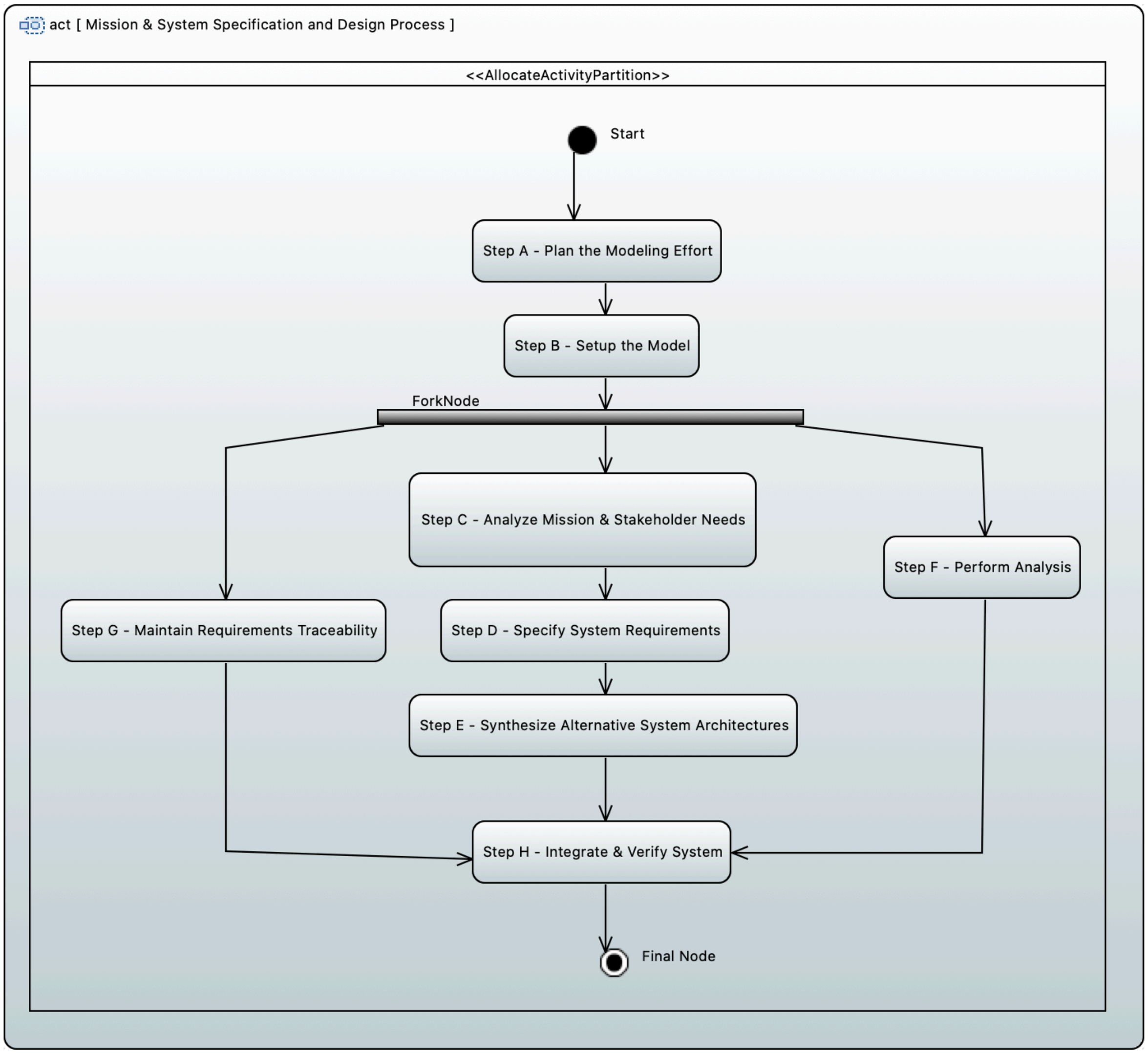


Figure 3: Activity Diagram - Mission and System Specification and Design Process [18]

In Step A, the system model incorporates initial planning activities such as objectives, scope for the modeling effort, MBSE method tailoring, schedule, roles and responsibilities, and the training approach. Step B shows how the model must be set up by defining organizational structure and conventions. Step C, analyzing the mission and stakeholder needs, defines the stakeholders, mission requirements, and elements that the system will interact with. In Step D, by specifying the system requirements, the launch pad system shall provide specific functions, interfaces, physical characteristics, performance qualities, and other quality characteristics. Synthesizing alternative system architectures, Step E, describes alternative configurations of system elements, and how the elements interact to satisfy the launch pad specifications. Step F, performance analysis of the launch pad system is completed throughout the development process to test alternative design configurations and ensure the design selected is the best fit. Step G is to maintain requirement traceability, and this is important to perform iteratively in order to verify the launch pad is meeting all stakeholder needs and managing the evolution of the design requirements. Finally, Step H integrates and verifies that the system is adopted early in the design process. This step helps to develop a cost-effective requirement verification approach and certifies the system will satisfy the intended mission. This may be performed by using simulation or analysis models.

After the hardware and software is implemented, integration and performance testing are completed to provide empirical datasets proving the requirements are fulfilled. The research of this paper generates models for launch pad subsystems in an iterative and recursive fashion with respect to Figure-3 steps A through H. The ISO/IEC-15288 technical process is selected as the system engineering framework, and MBSE is the medium utilizing SysML.

1.5 Thesis Framework

Chapter 2 discusses the highlights of the research performed, reviews common terminology in MBSE, and reviews popular model-based tools, languages, and methodologies. All nine of the SysML diagram types are discussed and crucial terms are covered prior to reviewing the diagrams for the launch pad. Chapter 3 capitalizes on the information gathered from Chapter 2 and builds MBSE diagrams using SysML for a launch pad. All nine SysML diagram types are built and tailored towards launch pad design. A framework tailored toward launch pads is introduced which leverages the advantages of MBSE. Chapter 4 reviews the architectural design process and outlines the next steps for launch pad MBSE design. Areas of improvement for the model are proposed. Lessons learned from model generation are covered. In Chapter 5, concluding remarks are made, and future MBSE design for launch pads is further discussed.

CHAPTER 2

2. LITERATURE DISCUSSION

2.1 A Brief History of Rocket Launch Sites

In 1957, the Soviet Union launched a modified SS-6 (Sapwood) carrying the first artificial satellite successfully placed into orbit from the spaceport known as Baikonur Cosmodrome. Since that time, twenty-seven spaceports across the globe have launched rockets to orbit. There are twenty-two active spaceports: 5 in the United States, 4 in China, 3 in Russia, 2 in Japan, and 1 in French Guiana, India, Iran, Israel, Kazakhstan, New Zealand, North Korea, and South Korea [41]. In the coming decades space launch frequency is projected to grow exponentially, and spaceport demand will grow to support this trend in aerospace. Sending mass into orbit is a difficult objective, which has fascinated many researchers and attracted engineers into the field who are hoping to improve the efficiency of rocket launches by exploring new paths to space.

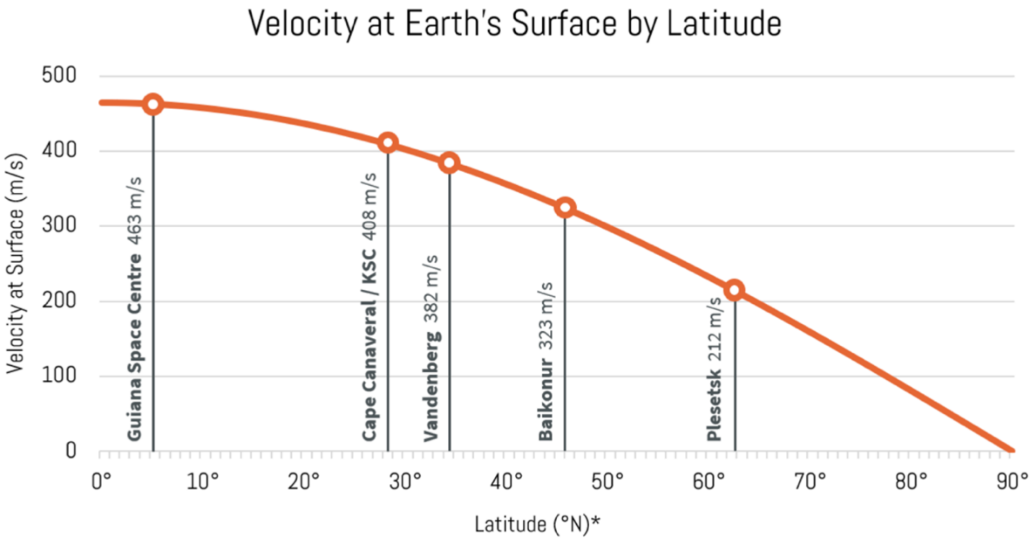


Figure 4: Velocity at Surface vs. Latitude for the 5 most active spaceports [41]

LauncherOne, a two-stage orbital vehicle under development by Virgin Orbit using RP-1 / LOX for their Newton rocket engines, is designing an air launch to orbit rocket for small payloads of 300 kilograms targeting a sun-synchronous orbit by releasing the launch vehicle from a Boeing 747-400 carrier [48]. The Northrop Grumman Pegasus, the first privately developed space launch vehicle, is an air-launched vehicle with three solid propellant stages and an optional fourth stage monopropellant that is released at approximately forty thousand feet designed to carry small payloads of up to one thousand pounds to LEO. The air launch to orbit technique dramatically reduces the spaceport's complexity by using a runway for horizontal takeoff but requires a custom aircraft to carry the rocket to altitude and release the vehicle for final ascent. Rocket payloads are then constrained by the aircraft's carrying capacity, limiting this technique to smaller payloads [38]. A multinational space launch service named Sea Launch designed a mobile maritime launch platform with the ability to park at the equator for launch. A rocket must deliver a satellite to 125 km for a circular orbit with adequate horizontal velocity of approximately 7 km/s for low Earth Orbit [41,51]. Due to the horizontal velocity requirement, the position of the spaceport on the Earth's surface leads some launch sites to be more optimal than others depending on the final orbit desired. In order to take complete advantage of the earth's rotation, a rocket has to be launched due east at the equator. Figure 4 shows the velocity at the Earth's surface (m/s) as a function of Latitude (ºN) for the five most active spaceports in the world.

Sea Launch took advantage of the Earth's maximum rotational movement of 465 m/s in the eastward direction at the equator to increase payload capacity [41]. The risk of a rocket failure near populated areas was eliminated with the maritime approach, and Sea Launch successfully delivered 32 rockets to a geostationary transfer orbit before the program was indefinitely suspended in 2014 [25].

While new methods for launching rockets are constantly being explored, approximately 99% of all orbital space launches to date use ground-based platforms [41]. In this thesis, research is strictly limited to ground-based spaceport design. Aside from the earth's rotation, other factors such as natural conditions (e.g. weather, environment), population density, azimuth limitations, political stability, accessibility, neighboring airspace, and public awareness are all considered when pinpointing future locations of ground-based spaceports. The five currently active (supported at least one orbital space launch over the past decade) spaceports in the United States are detailed below in Table-3.

Table 3: Active Spaceports in the United States of America [41]

Spaceport	Location	First Launch	Launches (1957-2018)
Cape Canaveral / KSC, FL	28.6ºN, 80.6ºW	2/1/1958	858
Vandenberg Air Force Base, CA	34.6ºN, 120.6ºW	2/28/1959	615
Wallops Flight Facility, VA	37.9ºN, 75.5ºW	2/16/1961	34
Ronald Reagan BMD Test Site, MH	9.1ºN, 167.7ºE	9/29/2008	2
Pacific Spaceport Complex, AK	57.4ºN, 152.3ºW	11/20/2010	2

The launch facilities at Cape Canaveral and the Kennedy Space Center are operated by NASA and the U.S. Air Force 45th Space Wing, offering a missile and rocket range that is over 10,000 miles from Florida to the Indian Ocean. Cape Canaveral spaceport also offers an FAA-licensed commercial launch site operated by Space Florida. This is the location of the USA's first launched satellite and is responsible for all of NASA's crewed space flights. The site specializes in low-inclination, prograde orbits, and northeast launches targeting the International Space Station. Launch vehicles at this site include the Shuttle, Delta II, Atlas, Delta-4, Falcon, Peacekeeper, and Falcon Heavy. KSC supports horizontally and vertically integrated rockets using mobile launcher platforms [41].

Vandenberg Air Force Base is the USA's second most active spaceport, recognized for its suitability to launch imagery satellites into orbit with high inclination, and two thirds of its' missions have final inclinations between 80º and 100º. The site has military and commercial customers, holding the record for the most successful polar-orbiting satellite missions. Rockets launched include the Delta II, Atlas, Peacekeeper, Delta-4, and Falcon [41]. SpaceX leases both launch pads at Space Launch Complex 4, using SLC-4E for the launch site and SLC-4W as a landing pad for the vertical take-off vertical landing (VTVL) Return-To-Launch-Site (RTLS) first-stage boosters onboard the reusable Falcon 9 [22]. In 2018, Firefly Aerospace announced that it plans to use Vandenberg's SLC-2W launch pad for future missions of their two-stage RP-1 / LOX small-satellite launch vehicle named Alpha, targeting up to 1000 kg payloads to LEO or 630 kg to a 500-kilometer SSO [17].

Wallops Flight Facility was founded in 1945 and is the longest-running rocket launching range in the United States of America. It features the FAA-licensed Mid-Atlantic Regional Spaceport (MARS) which operates the space launch facilities (PAD-0A, PAD-0B, and PAD-0C). With a higher latitude than KSC, MARS Launch Pad-0A is an ideal location to launch payloads to the International Space Station and currently supports the Northrop Grumman Antares rocket during cargo resupply missions to the ISS. Launch Pad-0B supports vertically integrated solid rockets such as the Minotaur, which first launched from MARS in December of 2006. Launch

Pad-0C is a new pad tailored to fuel Rocket Lab's Electron, containing Rutherford engines with the first electric-pump-fed engine. The two-stage small-lift launch vehicle is capable of lifting 300 kg to LEO [42]. No payloads originating from WFF have reached GEO or any inclination higher than 70º [41].

Ronald Reagan Ballistic Missile Defense Test Site is a U.S. Army-operated spaceport in an extremely remote location in the Marshall Islands. With an offset of 9º North from the equator, it is the southernmost U.S. operated facility. The site targets equatorial LEO missions. The Falcon 1 two-stage-to-orbit RP-1 / LOX rocket, owned by Space Exploration Technologies Corporation (SpaceX), became the first commercial company to use a liquid-fueled rocket to transport a satellite to orbit from this location on September 28, 2008 [31, 41].

The fifth and final active spaceport in the United States is the Pacific Spaceport Complex in Alaska. Formerly known as the Kodiak Launch Complex, this was the first FAA-licensed commercial spaceport built outside the vicinity of a federal test range. The commercial launch pad is tailored towards small solid-propellant launch vehicles. This is the northernmost spaceport in the United States, and while it is not ideal for launches requiring low inclination or high altitude, the geographic location efficiently launches payloads requiring sun synchronous (SSO) or polar orbit trajectories [2, 41].

Each spaceport is custom built and tailored to meet the needs of the launch service provider. Whether the customer is a commercial company (E.g. Northrop Grumman, SpaceX, Blue Origin, Rocket Lab, Firefly) or a military client, the requirement to build, operate, and maintain the spaceport in the most efficient manner to support upcoming launches is a shared desire at all facilities. In order to get to market at the fastest pace with a quality end result, spaceports require well-organized modeling tools with single sources of truth. As launch cadences become more frequent, spaceport development must quicken to support the trend.

2.2 System Engineering Models

A model in systems engineering is a logical, mathematical, or physical representation of a process, phenomenon, entity, or system that promotes and enables understanding [15]. The depth, fidelity, and breadth of the model will vary and be dependent upon the purpose. For example, a low fidelity geometric model created in CAD may display a three-dimensional geometrical layout that supports a trade study performed during the design phase. A high-fidelity analytical model could describe the dynamic behavior of a system utilizing a simulation to portray Navier-Stokes equations solving chemistry and flow simultaneously for a rocket engine. In this thesis, the system model is frequently used to describe logical relationships between a system's environment and the different elements of the system. An example of a system model would be a block diagram. This is neither a geometric nor analytical model, but the diagram easily explains to the user how components within a system are interconnected.

MBSE implies that the models are comprised of a united set of representations. These representations of system behavior and structure are interconnected and live within a central repository used by multiple engineers in parallel. The value of MBSE emerges from this central repository containing a collection of all the necessary system information. The repository encourages users to develop interconnected model elements and grants users the ability to effectively retrieve any desired information. In a document-based approach, this may require a meeting, phone call, or email in order to retrieve the latest design. The interconnected central repository also supports automatic propagation of changes to the design, identifies errors, and checks for consistency. These are the major benefits of an MBSE methodology.

2.3 SysML Overview

Technological advancements over the past few decades have supported MBSE's push towards object-oriented software concepts applied to systems engineering to support complex system development. In MBSE, the model is a part of a project. The project is a physical concept, a working unit that consists of the model and project configuration options. The model is a logical concept, an abstraction of a system which describes the structure in various aspects or viewpoints. The model consists of model elements such as requirements, use cases, actors, blocks, and one or more diagrams that show specific viewpoints of a system or combined systems. The coherent structure of a model can be achieved by utilizing packages. The package element in SysML can be used to create model hierarchy, and a package can contain model elements or other packages.

In this paper, the OMG Systems Modeling Language ™ (SysML ®), a graphical modeling language for system representation, will be explored. This language is intended to support systems engineers working on hardware, software, data, procedures, and facilities; building designs and specifications; and performing analysis and verifications. The Object Management Group (OMG) adopted the Unified Modeling Language and built the SysML extension which was formally released as SysML v1.0 in 2007. SysML is a modeling language and an international standard [21]. The current software release for SysML is 1.6 at the time of this writing, but as the OMG technology is adopted the software continues to evolve at a fairly rapid pace [46]. The following section reviews key SysML concepts in order for the reader to better understand the terminology applied in Chapter 3 to launch pad system models. SysML includes nine different diagram types as shown in Figure 5 [43].

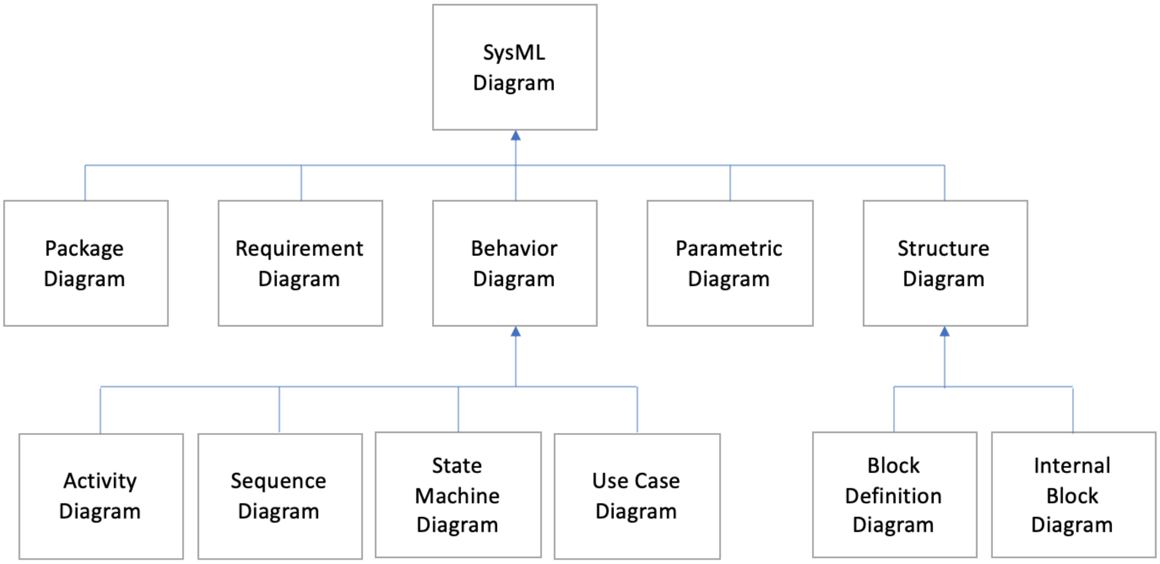


Figure 5: Nine Types of SysML Diagrams

In Figure 5, there are fields which are standard in all nine SysML diagram types. There is a diagram frame, content area, and diagram header, which provide the user with helpful information about the specific model being shown. The standard diagram header has four fields, shown in Table 4, but not all four of them are required, and this is customizable by the user [18].

Table 4: SysML Diagram Fields

Field	Description of field
Field #1	Identification of diagram type
Field #2	Model element kind
Field #3	Model element name that the frame represents
Field #4	Purpose of the diagram, title

2.3.1 Structure Diagrams

As shown in Figure 5, the two types of Structure Diagrams are the “Block Definition Diagram (BDD)” and the “Internal Block Diagram (IBD)”. A component, system, or external system are examples of a block, which is a general-purpose construct commonly used in SysML to represent any logical or physical unit. The block may take the form of software, hardware, a facility, datatype, or a user. A simple example of a block within a BDD is shown for a pressure transducer on a launch pad in Figure 6. Within this block, ‘Pressure_Transducer-O1’ is a block that has a range of 0-100 psig and performs an operation called ‘Report Ullage Pressure to Controller’. The block also has a port called ‘Controller_I/F’ that specifies the electrical interface to the launch pad control system. The block may be specified to have other functions, properties, interfaces, and features.

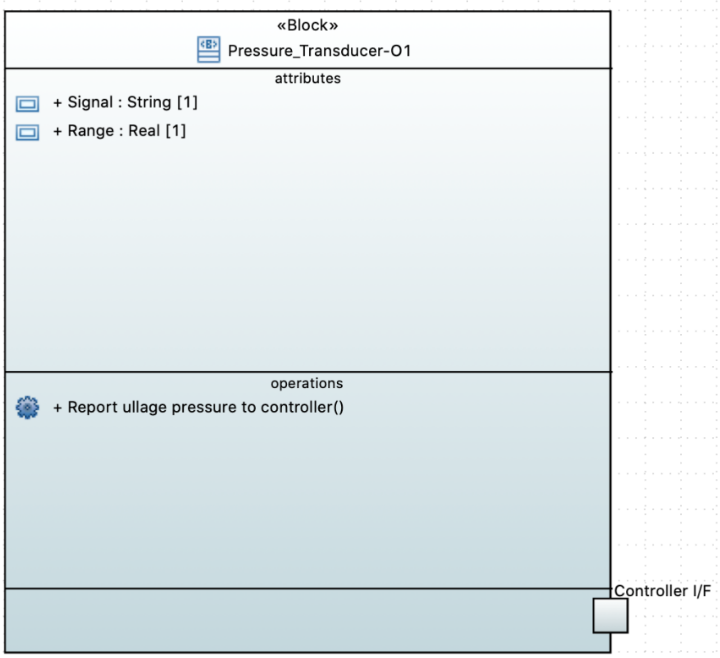


Figure 6: Example of a Block in SysML

The primary purpose of a Block Definition Diagram is to define the blocks and relate blocks to one another through relationship types such as reference, generalization/specialization, connection, or whole-part relationships. A black diamond describes the "whole-end," and the arrow designates the subsystem or 'part' [18]. In Figure 7, the launch pad 'system' is at the top of the tree, with six major subsystems which are mechanical, controls, fluid ground support equipment, electrical power system, hydraulics, and safety. The fluid ground support equipment subsystem is further decomposed into the ECS, oxidizer, fuel, water deluge, and gases. The number above each subsystem is the block's multiplicity, or quantity. If there is no number, the default multiplicity is one. The multiplicity may also have a lower and upper bound, shown on the hydraulic subsystem as '0..1' in this example, meaning the system is optionally included as part of the whole. It may be useful to aggregate a set of components into a logical whole relationship, or the component may be owned by another whole-part relationship.

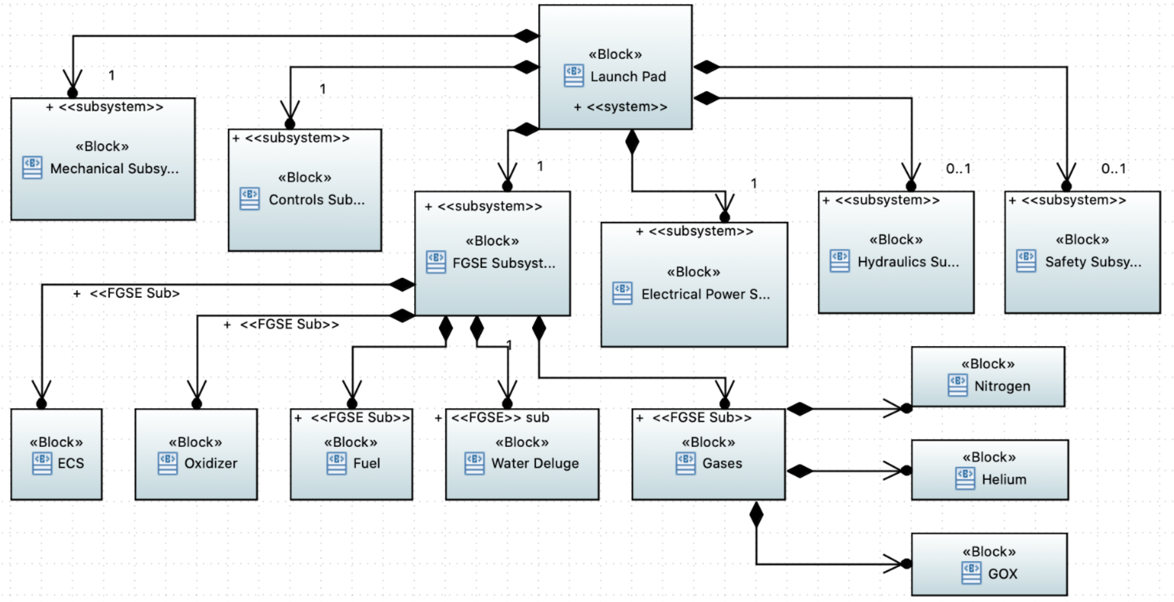


Figure 7: Launch Pad Overview (BDD)

The generalization or specialization relationship between blocks in SysML is used when a general block such as the FGSE subsystem has properties, functions, or interfaces, which a more specialized block inherits. The more specialized block, such as the FGSE subsystem titled ‘Oxidizer’, will then display the more unique features to avoid defining common qualities for each specialization. This promotes reuse.

The second type of structural diagram in SysML is an internal block diagram (IBD) represented in Figure 8. The main purpose of an internal block diagram is to represent the connections between different systems, subsystems and components. The following example shows flow and connection points of the electrical power system from the facility to a controller on the launch pad. This controller may be referred to as a part, which is a general name for an interconnected component. A connector may be used to connect ports on specific parts or to connect two parts together directly. A port is a location on a block which serves as an interaction point, enabling interfaces to be displayed [47].

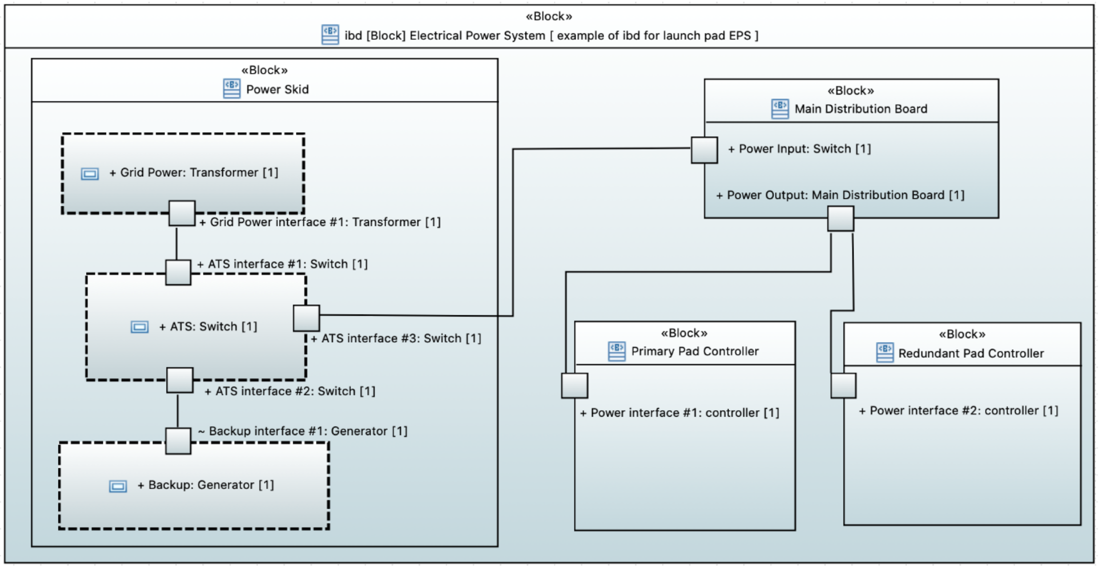


Figure 8: EPS Connection Example (IBD)

There is an important difference between parts and blocks which the modeler must understand. An entity's specific definition is a block, while the use case of an entity within certain context should be classified as a part. For example, refer to the figure below depicting the flow control skid for the fueling subsystem of the launch pad with a primary and backup flow control valve. Each of these flow control valves are identical copies of one another with one key difference; one is the primary and second is the redundant. The definition of these two pieces of hardware is equal, but in the context of the fueling system they have different use cases or roles. The primary flow control valve and the backup flow control valve are parts defined by the same block, and the "Control Valve" is the block (i.e. definition). Along with the primary and backup flow control valves, the flow control skid example below consists of an inlet pressure sensor, a check valve, and an inline filter with an attribute showcasing the filters rating of one hundred microns.

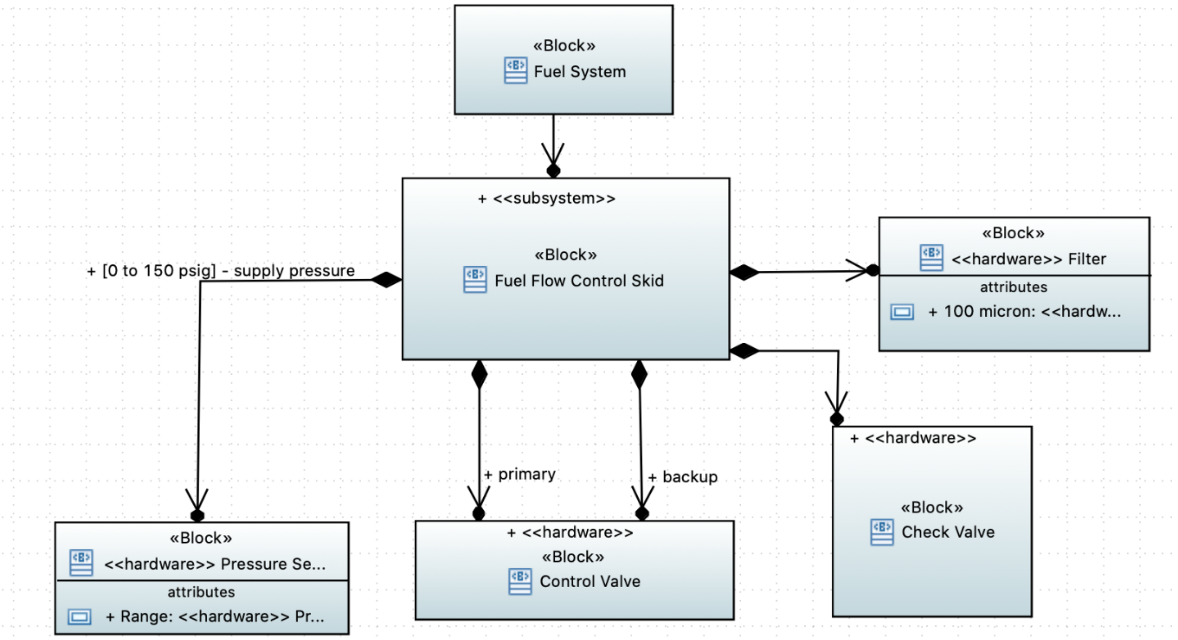


Figure 9: Fuel Flow Control Redundancy Example (BDD and IBD)

2.3.2 Behavior Diagrams

The four types of system model behavior diagrams in SysML are the activity diagram (ACT), sequence diagram (SD), state machine diagram (STM), and use case diagram (UC). Table 5 lists some of the key elements which appear in modeling software for the four behavior diagram types [47].

Table 5: Behavior Diagram Key Elements [47]

ACT - Activity	SD – Sequence	STM – State Machine	UC – Use Case
Structured Activity	Actor	State	Actor
Action	Lifeline	State Machine	Use Case
Action Pin	Boundary	Initial	Test Case
Partition	Control	Final	Collaboration
Control Operator	Entity	History	Collaboration Use
Parameter	Fragment	Synch	Boundary
Object Node	Endpoint	Object	Package
Central Buffer Node	Diagram Gate	Choice / Junction	Use
Data Store	State	Entry / Exit	Associate
Decision	Continuation	Terminate	Generalize
Merge / Synch	Interaction	Fork/Join	Include
Initial / Final Flow	Message	Transition	Extend
Region	Self-Message	Object Flow	Realize
Exception	Recursion	Event / Signal	Invokes
Fork/Join	Call	Trigger	Precedes

The SysML Activity Diagram (ACT), an extension of the UML Activity Diagram, is a powerful tool which is used to model control flow, inputs, and outputs. A sequence of actions, which may contain input and output pins, is symbolized by the behavior of blocks using control flows. Depending on the system and activity, these items could be hardware, energy, data, power, information, and anything else the modeler requires producing, consuming, or conveying.

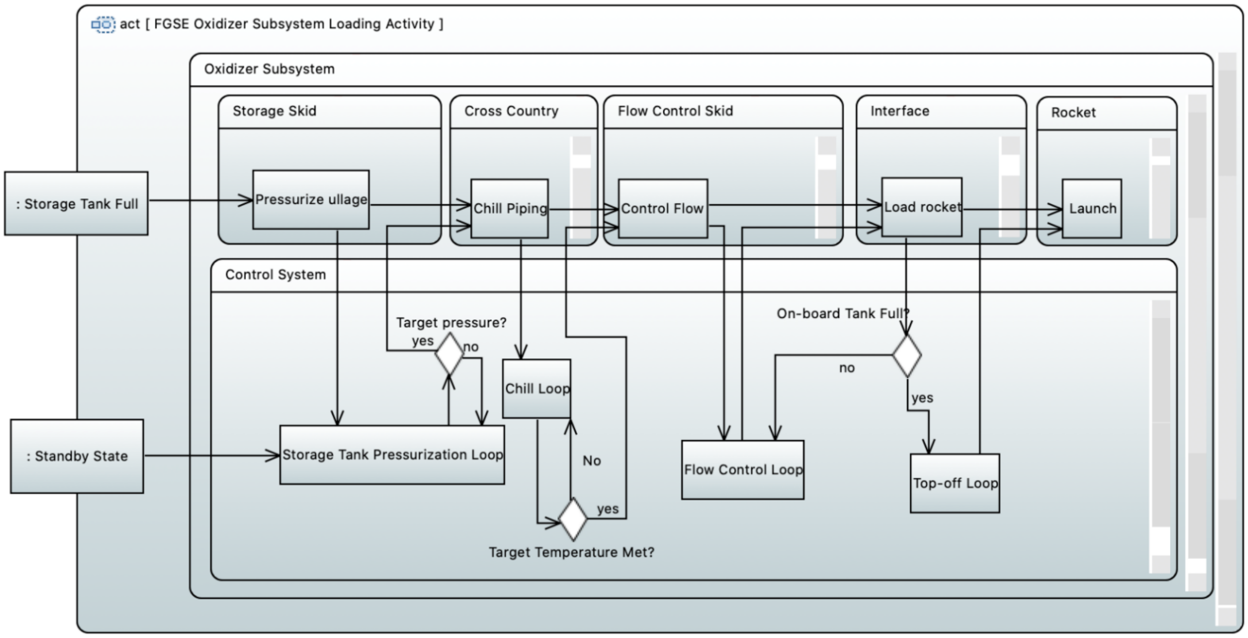


Figure 10: FGSE Oxidizer Subsystem Loading Example (ACT)

An activity diagram is an appropriate place to define instances where parallel processing may occur during system performance. Activity partitions may be created with the use of swim lanes, organizing the control flow whether the activities are in series or parallel. Object flow is a term used to portray how the output of one action interconnects to the input of a second action. Figure 10 depicts a simplified FGSE version of an activity diagram for the oxidizer subsystem loading commodity from the launch pad storage area to the launch vehicle interface. The frame on the exterior of the model embodies the activity which contains actions. Every action is performed by a component of the FGSE Oxidizer Subsystem with activity partitions (swim lanes) to help organize the flow. Execution of specific activities is reliant upon the flow of tokens. One action may not begin until the predecessors or input tokens are available on the block's input. For example, the control system does not enter the chill loop until the ground instrumentation records target pressure in the ullage space of the storage tank. By aligning predecessors and successors via tokens, the modeler may precisely identify the expected system's behavior based on the consumption and production of tokens.

In order to control tokens throughout an activity, many types of control nodes are implemented. The activity begins execution when a token is available on the initial node. The activity terminates when a token arrives at the activity final node. In between the initial node and activity final node, other control nodes such as decision nodes and join nodes help control the activity flow. A decision node is modeled with a diamond, which controls the path of a token based on the resultant of a guard condition. A join node contains two inputs and will not execute until tokens have landed on both of the input ports. The nodes communicate via specialized action types. For example, signals may be sent from one node to another using a send signal action. Signals may be received on an input to the node using an accept event action. The call behavior action calls upon a separate activity, allowing the modeler to link multiple activities and run nodes in series or parallel. The execution of the activity is controlled through the sending and receiving of the signals or tokens which translate into real events in the system being designed. Other features the modeler can apply to precisely design activities include time-continuous inputs and outputs, streaming, interruption of actions based on the arrival of a token, and many more [43]. The second type of SysML behavior diagram utilized in the launch pad system model is the Sequence Diagram. The Sequence Diagram is abbreviated (SD) in the diagram frame, and its purpose is to describe system behavior through the usage of a sequence of messages that are communicated between different parts (or lifelines) of the system [18]. Another use case is to represent timelines of specific events, such as the pressurization of an oxidizer storage tank. The sequence diagram in Figure-11 represents the operation, beginning with the system engineer and technician configuring the storage tank manual valves and recording starting readings on instrumentation. Then, the operator starts the press loop, interfacing with the control system which acknowledges a set point and automatically commands valves open and closed to maintain ullage pressure. When the operator sends the stop command, the pressurization loop vents the ullage pressure to standby conditions; then the technicians secure the field components, and a final tank level reading is recorded to determine the amount of commodity used.

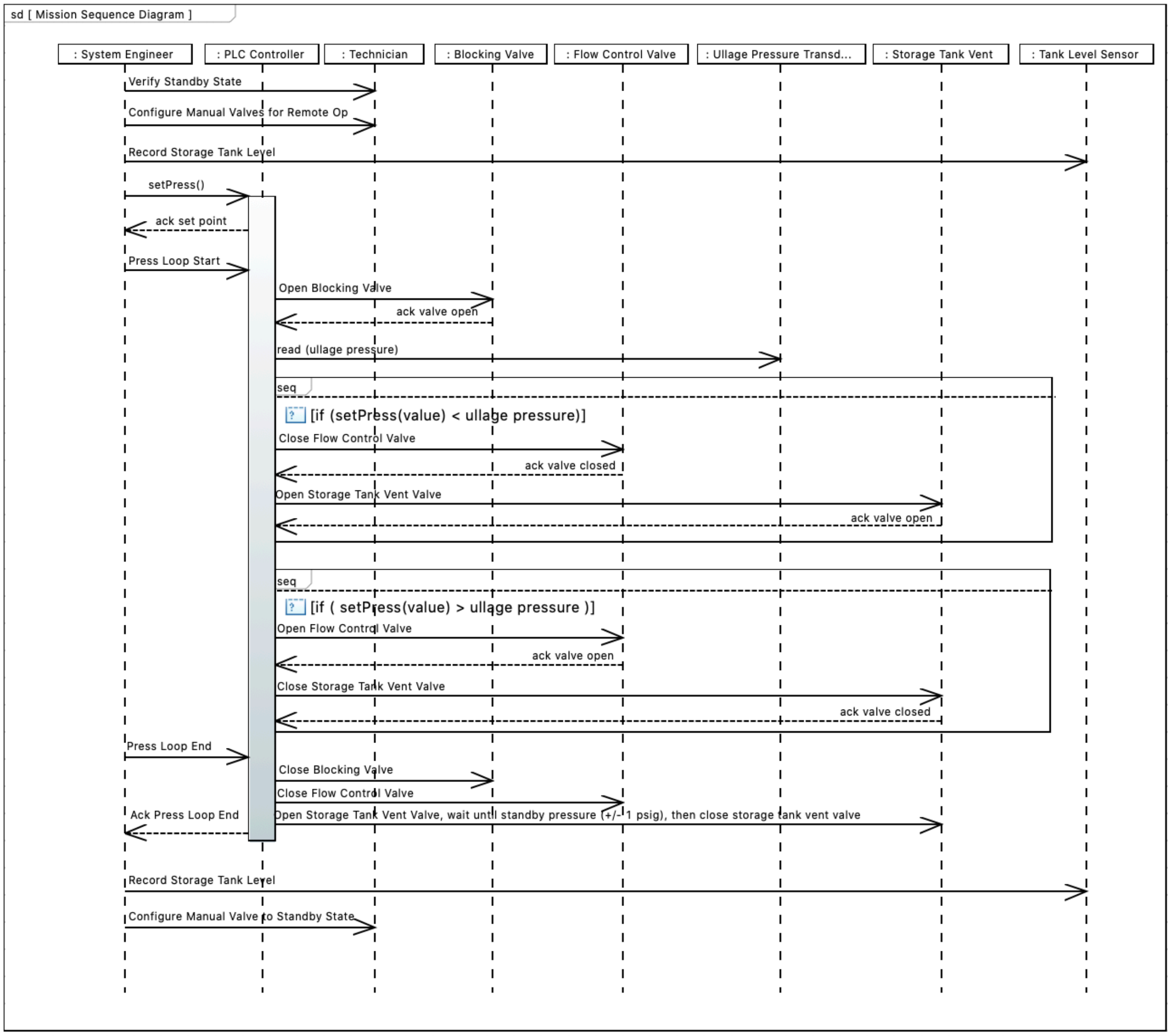


Figure 11: Oxidizer Pressurization Loop Example (SD)

In Figure 11, lifelines are used for the system engineer (actor), PLC controller, technician (actor), blocking valve, flow control valve, ullage pressure sensor, storage tank vent, and tank level sensor. Blocks at the top of the figure are referred to as parts of lifelines. The lines with arrowheads signify messages being sent between different parts of the sequence diagram. The default configuration is for time to advance as the modeler proceeds down the vertical axis.

The third type of behavior diagram is the State Machine Diagram, abbreviated as (STM). This diagram type is heavily used when modeling important phases or discrete states of an element, as well as transitions from one specified state to another. The state machine is applied to technical use cases of blocks [43]. A state machine diagram is used to depict the condition of a block, such as the open state or closed state of an electro-pneumatic control valve on a launch pad. The transition between the open state and the closed state of a control valve is triggered when a solenoid is either energized or deenergized. Once the discrete output signal from the PLC is sent to the end device, a 24 VDC solenoid energizes allowing pneumatic pressure to flow to a spring-return actuator which compresses internal springs and causes a ball valve to rotate ninety degrees. Once the ball valve has cycled, the flow path is either opened or closed depending upon the standby configuration of the ball valve coupled to the actuator and valve controller. A simple example of a state machine diagram (STM) for an electropneumatic control valve is shown within Figure 12. Within the state machine diagram, exit, entry, and do behavior types may also be defined by the user to further explain the purpose of a block while in a specified state. In the state machine diagram example, the control valve is normally closed. When the solenoid is energized, the valve turns to the open state. When the valve's feedback, a discrete input to the controller, registers as true, a 'do' behavior is started telling the fueling system that there is an open path allowing commodity to flow past this gate. In Figure 12, the STM's 'do' behavior is titled ‘Flow Fuel to Flow Control System’. The diagram gets more complicated when specific system behavior is tied to transitions between states. As an example, once the control valve in Figure 12 begins transitioning from the closed state to the open state, the flow control valves downstream may acknowledge this state and begin another STM (not pictured) to properly control flow based on a commanded setpoint. SysML makes it possible to integrate state machine diagrams, activity diagrams, and sequence diagrams to highlight critical characteristics of a system’s behavior.

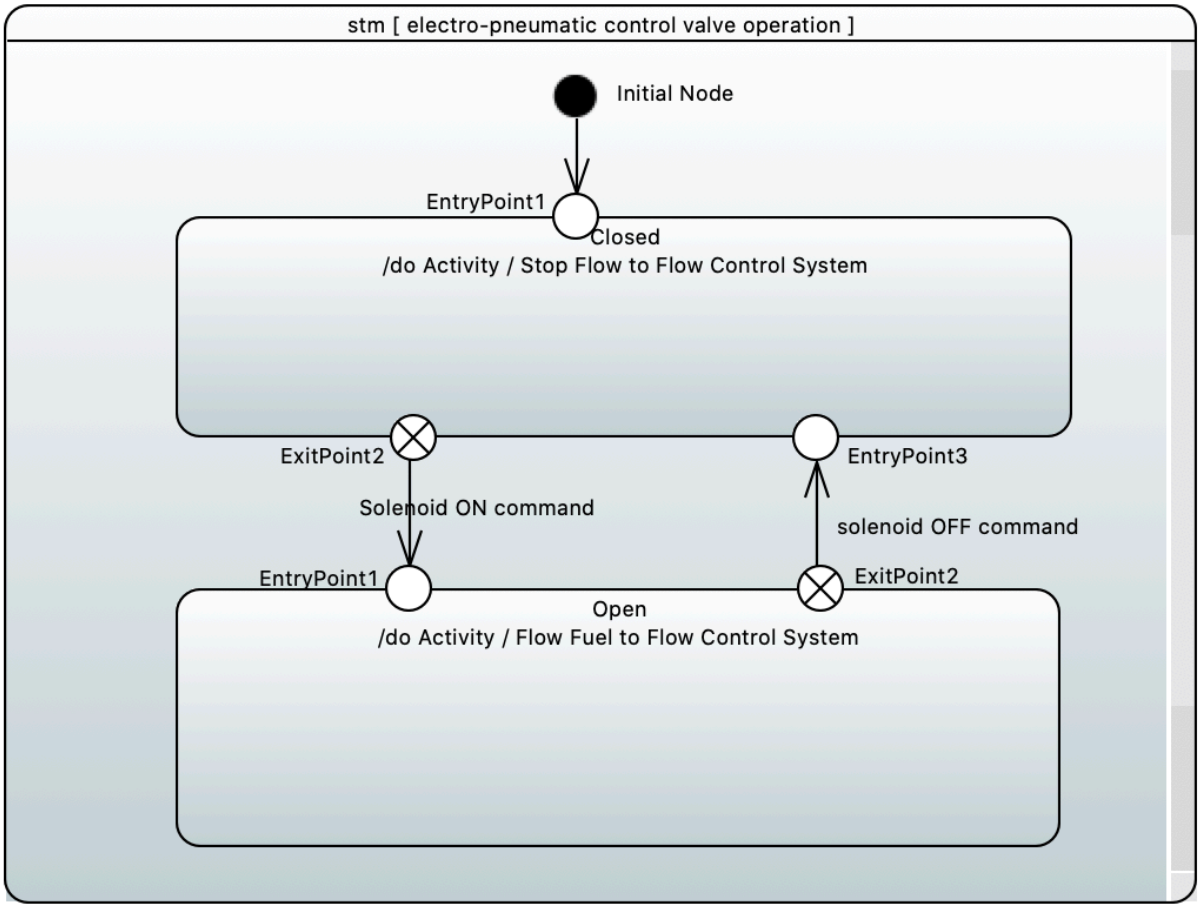


Figure 12: States of an Electro-Pneumatic Control Valve (STM)

The final type of behavior diagram used in SysML is titled the Use Case Diagram (UC). This type is primarily focused on describing the goals, requirements, or mission objectives of the system. The key elements of a UC diagram are the use case, the subject, and the actors. The use case is the goal or main objective, the subject is the system at hand, and the actors are external systems that interface with the primary system to achieve the mission [18]. An example of a use case diagram is one involving a camera system on a launch pad. The actors in the diagram include the *launch service provider*, *NASA Safety Engineer*, *Launch Pad Camera Operator*, and *Fire Department*. A main goal of the *Ground Control Operator* is to *Provide camera views of Fuel and Oxidizer Systems while Loading the Launch Vehicle* to the NASA Safety Engineer and the launch service provider. This promotes the much broader goal of safely fueling the launch vehicle prior to flight and is represented by the <<include>> relationship in Figure 13 giving an example of a Use Case Diagram.

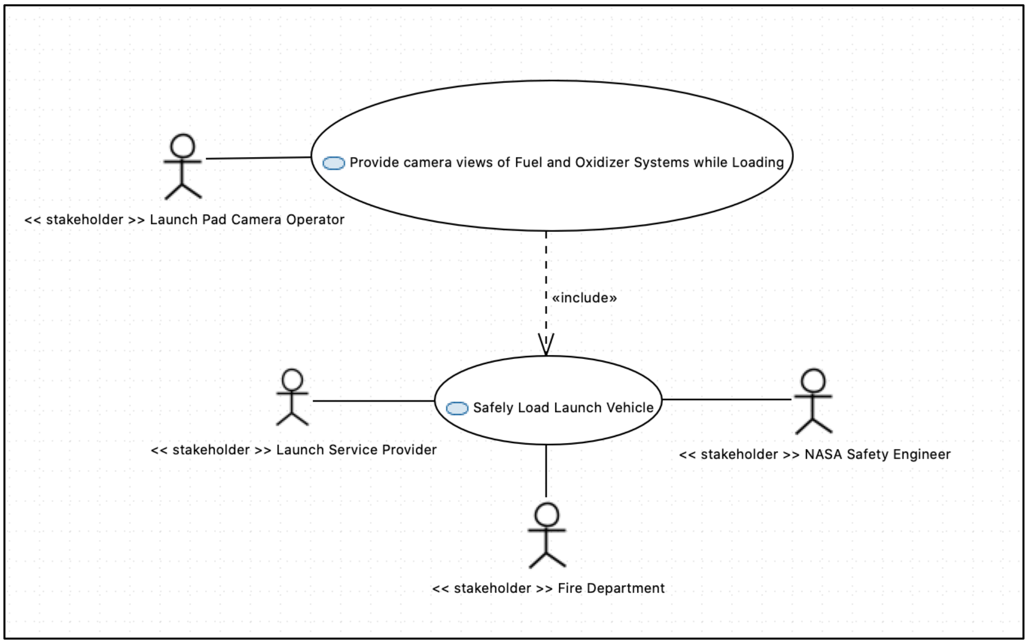


Figure 13: Cameras Monitoring Vehicle Loading Example (UC)

2.3.3 Constraint Modeling

So far, two types of structure diagrams (BDD, IBD) and four types of behavior diagrams (SD, ACT, STM, UC) have been covered with examples of each type. The next type of SysML diagram's purpose is to enforce mathematical rules with respect to block value properties, and they are referred to as parametric diagrams (PAR). The key element of the parametric diagram is the constraint block, which is shown as a rectangle with a keyword in double brackets. The constraint block defines a mathematical rule with parameters which are bound to block value properties, propagating to other block value properties in accordance with the mathematical rules set by the block. As shown earlier, blocks may contain value properties (e.g. mass, size, reliability, cost) or constraints such as Ohm's Law. The parametric diagram is a specialized version of the IBD which enforces constraints defined by specific blocks that are defined by block parameters. When implemented properly, the modeler may utilize BDDs, IBDs, and PARS in applications that are recursively scalable and capable of simulation [43].

The parametric diagram in Figure 14 is modeling a constraint called 'Valve Sizing' displaying the input and output parameters which are designated as squares on the edges of the constraint inner boundaries. The objective of this parametric model is to properly size a Class 300 globe valve with a 3-inch valve size. When the plant initially started up, the valve was not operating at the maximum designed capability. The system is sized for maximum expected operating pressure, but the customer has a desire to install a control valve sized for current operating requirements. Concentric reducers are installed in line with the valve, which has an upstream line size of 8 inches. The first step in the parametric diagram is to specify the variables necessary to size the valve. To solve, inputs required include the standard volumetric flow rate (q), upstream pressure (P1), downstream pressure (P2), line temperature (T1), liquid specific gravity at the inlet (ratio of liquid density at flowing temperature to density of water at 60ºF, dimensionless), absolute vapor pressure of the liquid at the inlet temperature (Pv), and the absolute thermodynamic critical pressure (Pc). Equation constants (N1 and N2) are determined from valve sizing lookup tables in the Emerson Control Valve Handbook [11]. Derived measurements are calculated for the piping geometry factor (Fp) and delta pressure (ΔP). The proper pressure drop value must be determined to properly size the valve. It is determined that the actual pressure drop is lower than the choked pressure drop, driving the ΔP to be equal to P1 minus P2. Once all of the derived measurements are calculated, the valve sizing coefficient (Cv) is found using the inputs (q, ΔP, N1, Fp, and density) to the equation on the right side of the parametric diagram shown below.

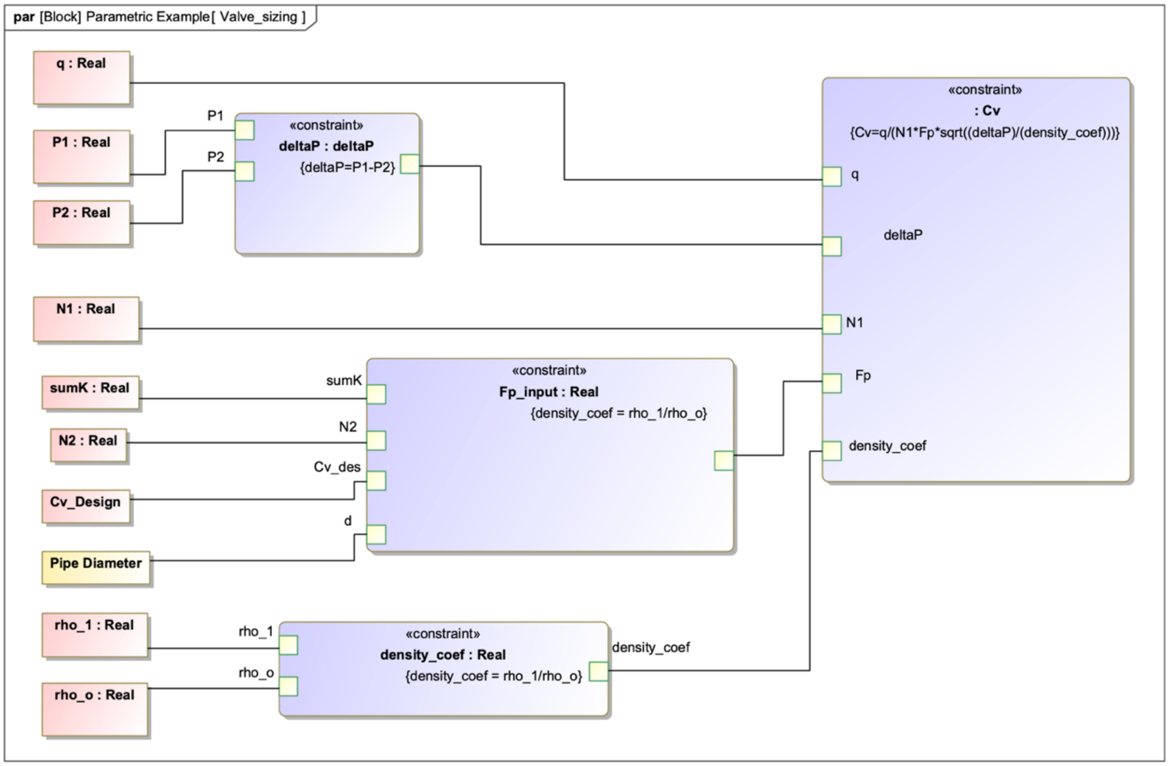


Figure 14: Control Valve Sizing Calculation (PAR)

2.3.4 Requirements

Per the SysML specification, “a requirement specifies a capability or condition that must (or should) be satisfied, a function that a system must perform, or a performance condition a system must achieve” [36]. In systems engineering, it is common practice to combine similar requirements into a specification. A typical challenge is writing the requirements in a way so that there are no contradictions, the requests are feasible – sufficiently written to ensure customers’ needs are validated – and verified to guarantee the system design and final product satisfy the requirements.

In traditional document-based systems engineering, requirements are traced back to a spreadsheet or text file which identifies an owner for each system requirement. It is then the responsibility of the requirement owner, typically a system engineer who is a subject matter expert (SME), to then design the system, run the analysis, and provide the results to the management team. Requirements can be extensive and often change after the original documentation is released for design, making it difficult for customers, managers, and other SMEs to keep track of all mission requirements and how these designs affect other systems as the design evolves. Issues arise as designs conflict. For example, a high-pressure nitrogen system may originally require remote monitoring of a purge to a vehicle interface using a pressure transducer. Months later after construction has started, the customer requests a redundant method of monitoring the pressure at this specific interface as well as a method to remotely turn off supply. This leads the pneumatics SME to add a pressure transducer and remotely controlled valve to the design. The control system was designed to support eight analog input channels in the high-pressure nitrogen system and sixteen discrete outputs and inputs. Unfortunately, all of the analog inputs and discrete inputs and outputs linked to this PLC have already been taken by other hardware. The controller for this system was already fabricated without accounting for the required expansion channels. This new requirement drives the controls engineer to make last minute changes to their design in order to account for the delta, driving cost due to the additional hardware, software implementation, and schedule due to lead times on parts required. If a SysML system level model is properly designed, the engineering team has the ability to link requirements to all coexisting systems and reduce the element of surprise.

In SysML, a requirements diagram is a tool to represent the typical document-based artifacts as a tree of requirements organized in an efficient way to promote visibility for all users. The diagram is particularly useful to visually represent hierarchy of specifications. Requirements within SysML use the relationships defined as either satisfaction, derivation, verification, refinement, or trace [18]. An example of a requirements diagram is shown in Figure-15 describing a portion of a launch pad's nitrogen high-pressure purge requirement. Within the model-based method, the requirement is satisfied by hardware blocks which include two pressure transducers, a discrete control valve to open or close the valve leading to the vehicle interface, and the I/O which ties to the controller for remote monitoring capability as a derived requirement. The traceability of a requirement to a system design is also shown within Figure 15. The sensing requirement for the high-pressure nitrogen purge for the GN2 Purge sensor specification specifies the required accuracy and range. The requirement is fulfilled via the accuracy and range of the HP GN2 purge sensor block and is indicated by the <<satisfy>> relationship. These values are derived from the "launch pad specification" and the "functional performance requirements" for "pneumatic sensor hardware" [18]. There are test cases depicted which verify the functional and performance requirements of the high-pressure nitrogen system.

These test cases will report true if the accuracy, range, and remote-control requirements are able to be met with the system designed.

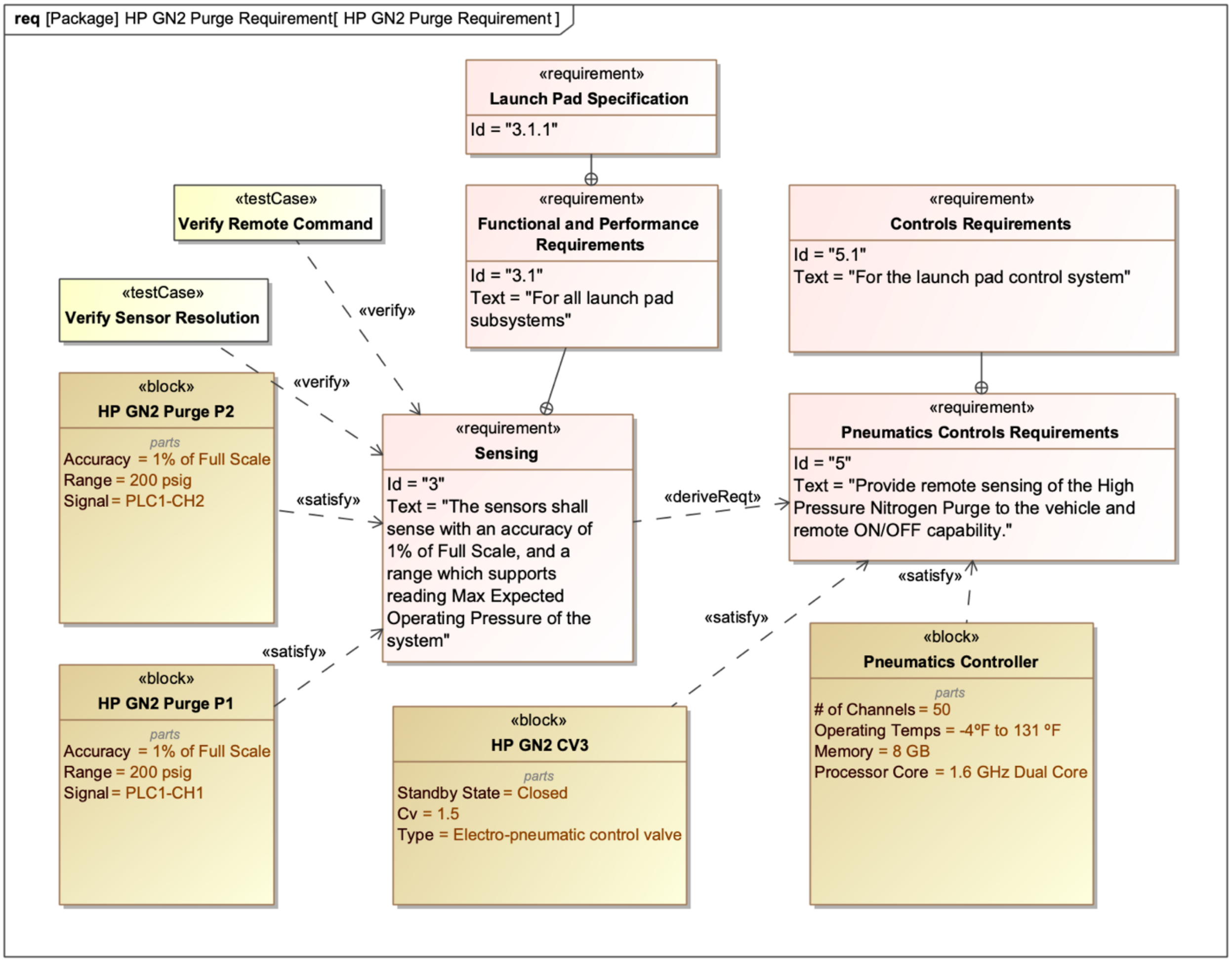


Figure 15: High-Pressure Nitrogen Purge Requirement Example (REQ)

2.3.5 Package Diagram

The SysML model is a complex structure which has the potential to include millions of model elements. The model elements may represent a system, component, component feature (property, interface, function, or relationship), or something else that is expressible in SysML.

Each model element is enclosed within a container referred to as the model element's parent or owner. In SysML, this model element that is enclosed is called the child element. Child elements may also be containers, giving the designer the ability to nest containment hierarchy of model elements. The system level model needs to be managed and properly organized in order to facilitate efficient use and retrieve information at the fastest rate possible. This is handled by using packages, which are a type of container for model elements. A package is similar to a directory structure on a computer, which provides a simple way to organize artifacts into logical groupings. Packages may be comprised of parametric elements, structures, behaviors, requirements, and other SysML model elements. Effective model organization makes it easier to reuse packages, improves accessibility and navigability, and aids organizations with configuration management of the model as well as information exchange with other software programs [43].

A package diagram (PKG) shows the model elements which are contained within a certain package. The package diagram's principal use case is to describe a model's organizational structure and to define SysML profiles or language extensions. The package has a unique name and an optional URI, which makes the object web accessible. In SysML, a model is the top-level package in the nested hierarchy. A model library is a specific type of package which is created with the purpose of reusing the elements contained within the library. On a launch pad, for example, components are typically reused in multiple systems in order to reinforce redundancy and have multiple spare pieces of hardware that may be installed in various locations. A model library of components that includes relief valves, control valves, ball valves, flow meters, pressure transducers, temperature sensors, and other hardware used on a launch pad could be reused and accessible by multiple system engineers. An example of a package diagram is shown in Figure 16. The content area within the package diagram displays multiple packages and elements within the package that are categorized by the frame. A folder symbol is the common symbol to denote a package. A URI, if used, will commonly appear within braces following the name of the package. Within this high-level controls package diagram, the subject matter experts for this system may begin to organize artifacts. Once the documentation and requirements are well understood, it is the system engineer's responsibility to trace every component to a controls requirement and verify the proposed design will meet the needs of the customer [43].

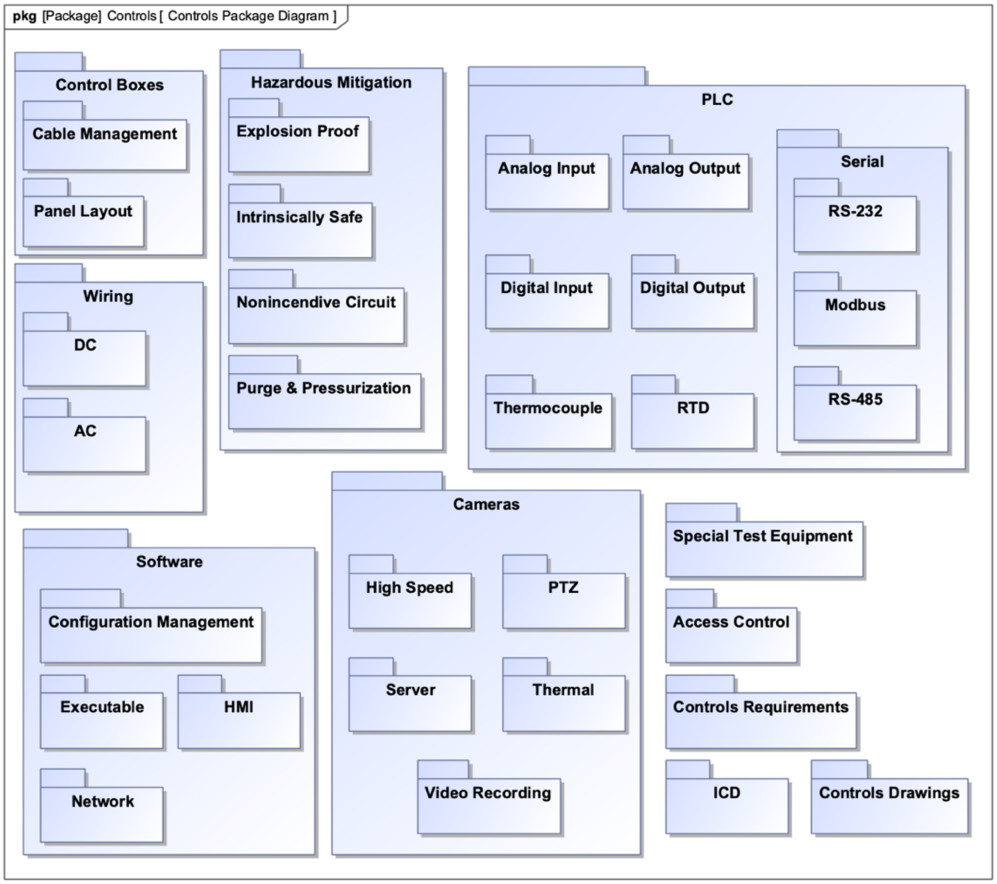


Figure 16: Controls Subsystems Packages Example (PKG)

CHAPTER 3

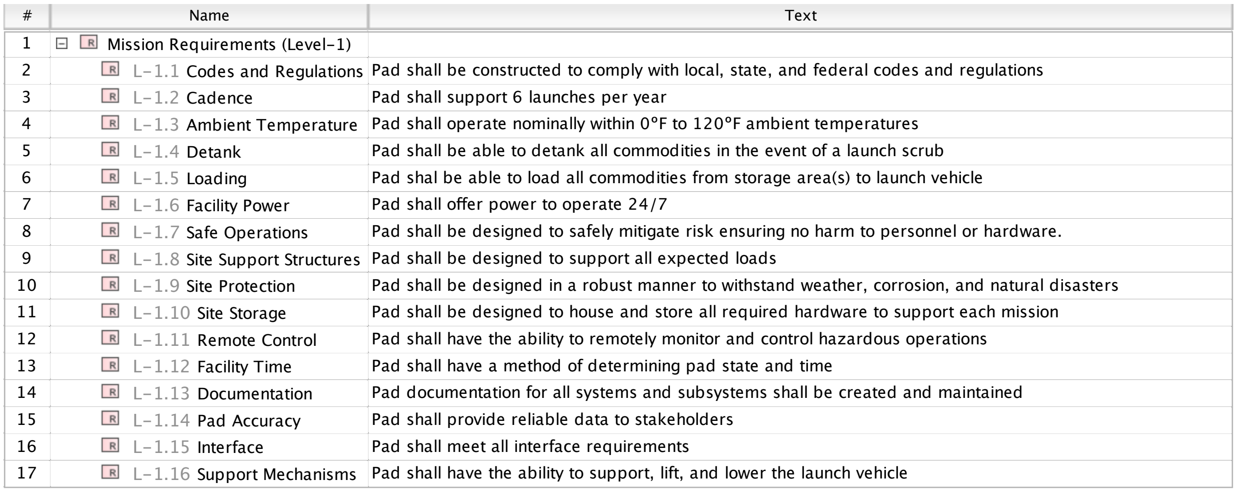
3. REVIEW OF THE LAUNCH PAD MODEL USING SYSML

The technical process outlined by ISO/IEC 15288 is selected as the framework to be followed for this MBSE application. The framework helps to establish and update work plans, assess progress with respect to requirements, guide project decisions, manage risk, and help to capture, store, and disseminate information to the project team. The technical processes span all life cycle stages. The technical processes include stakeholder requirements definition, requirements analysis, architectural design, implementation, integration, verification, transition, validation, operation, maintenance, and disposal.

3.1 Stakeholder Requirements Definition and Analysis Process

The first step in the ISO/IEC 15288 technical process is to define stake holder requirements and analyze them. The purpose of stakeholder requirement definition is to produce, document, and maintain stakeholders' needs regarding a system-of-interest. Inputs for requirement definition involve a description of the stakeholders' needs, timeline, budget, constraints, terms, conditions, and industry standards or specifications. The output of the stakeholder requirement definition process is a formally documented and accepted set of requirements which will govern the project. The purpose of requirement analysis is to examine, evaluate, prioritize, and balance all documented requirements and transform this formal document into a functional or technical view of the system-of-interest [49].

The key stakeholders for this fictional launch pad include the launch service provider (LSP), the LSP's end customer (satellite company), NASA, and the contractor designing and building the launch pad. The fundamental problem to address for this mission is the customer's need to build a launch pad within two years in order to support rocket launches at a cadence of six times per year with the end goal of deploying small-class payloads into space. A plan must be developed to achieve the objectives within the given constraints. The first step is to define the mission level requirements (Level-1). A first iteration of the mission level requirements is outlined in Figure 17. An example of a Level-1 requirement originating from the needs of the LSP is to launch a rocket from the pad every six months. For NASA, a Level-1 requirement is the necessity to construct the site while following applicable codes and regulations to ensure a safe working environment. For the payload customer, having a constant power source at the launch pad interface is a mission-level requirement in order to perform pre-launch testing.



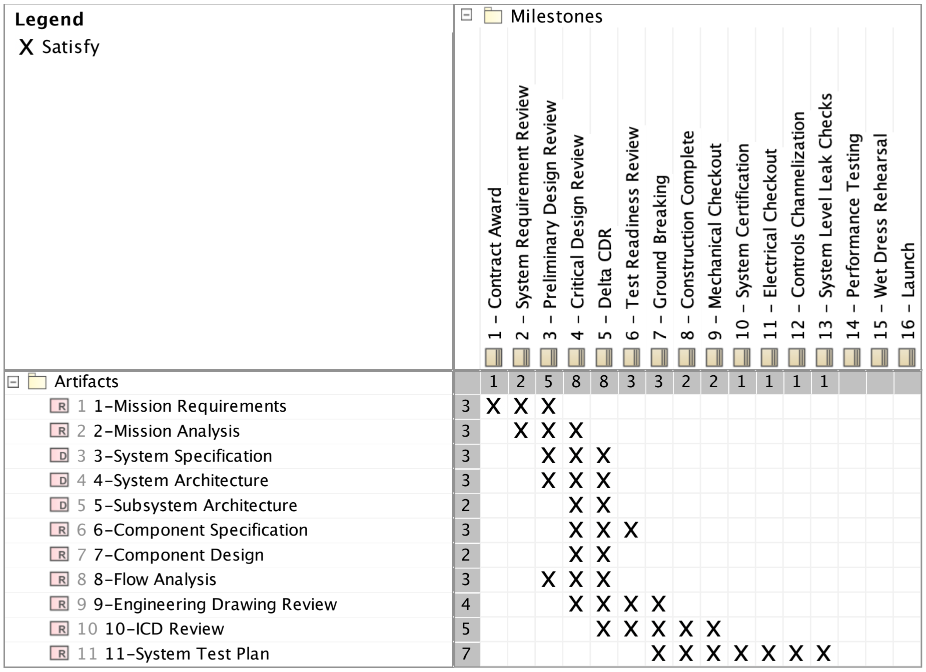
#	Name	Text
1	Mission Requirements (Level-1)	
2	L-1.1 Codes and Regulations	Pad shall be constructed to comply with local, state, and federal codes and regulations
3	L-1.2 Cadence	Pad shall support 6 launches per year
4	L-1.3 Ambient Temperature	Pad shall operate nominally within 0°F to 120°F ambient temperatures
5	L-1.4 Detank	Pad shall be able to detank all commodities in the event of a launch scrub
6	L-1.5 Loading	Pad shal be able to load all commodities from storage area(s) to launch vehicle
7	L-1.6 Facility Power	Pad shall offer power to operate 24/7
8	L-1.7 Safe Operations	Pad shall be designed to safely mitigate risk ensuring no harm to personnel or hardware.
9	L-1.8 Site Support Structures	Pad shall be designed to support all expected loads
10	L-1.9 Site Protection	Pad shall be designed in a robust manner to withstand weather, corrosion, and natural disasters
11	L-1.10 Site Storage	Pad shall be designed to house and store all required hardware to support each mission
12	L-1.11 Remote Control	Pad shall have the ability to remotely monitor and control hazardous operations
13	L-1.12 Facility Time	Pad shall have a method of determining pad state and time
14	L-1.13 Documentation	Pad documentation for all systems and subsystems shall be created and maintained
15	L-1.14 Pad Accuracy	Pad shall provide reliable data to stakeholders
16	L-1.15 Interface	Pad shall meet all interface requirements
17	L-1.16 Support Mechanisms	Pad shall have the ability to support, lift, and lower the launch vehicle

Figure 17: Mission Level-1 Requirements

Once mission requirements are captured, the next steps are to determine the model's objectives and scope, identify the most important milestones and deliverables, select a modeling method, and choose the proper software toolset. The objective of this launch pad model is to develop an MBSE-based description of the site's architecture that spans all required disciplines and subsystem views (civil, controls, FGSE, safety, EPS, and mechanical) while verifying a concise and well-understood flow down of the mission, system, and subsystem requirements. In the first iteration, the model concentrates on accurately defining mission scenarios, identifying system design options, developing a clear package structure, and ensuring the Level-1 requirements can be fulfilled. In subsequent iterations, the detailed block properties, component interfaces, comprehensive software logic, and other system characteristics shall be refined.

The model artifacts required to support the project milestones include the mission requirements, mission analysis, system specification and architecture, component specification and design, flow analyses, engineering drawings, interface control document (ICD), and system test plans. The milestones linked to these artifacts contain, but are not limited to, the contract award, requirement review, design reviews, construction start/end, mechanical checkout, leak checks, channelization, performance testing, verification, and validation. The matrix in Table 6 summarizes these artifacts with respect to the project milestones.

Table 6: Modeling Artifacts Versus Milestones



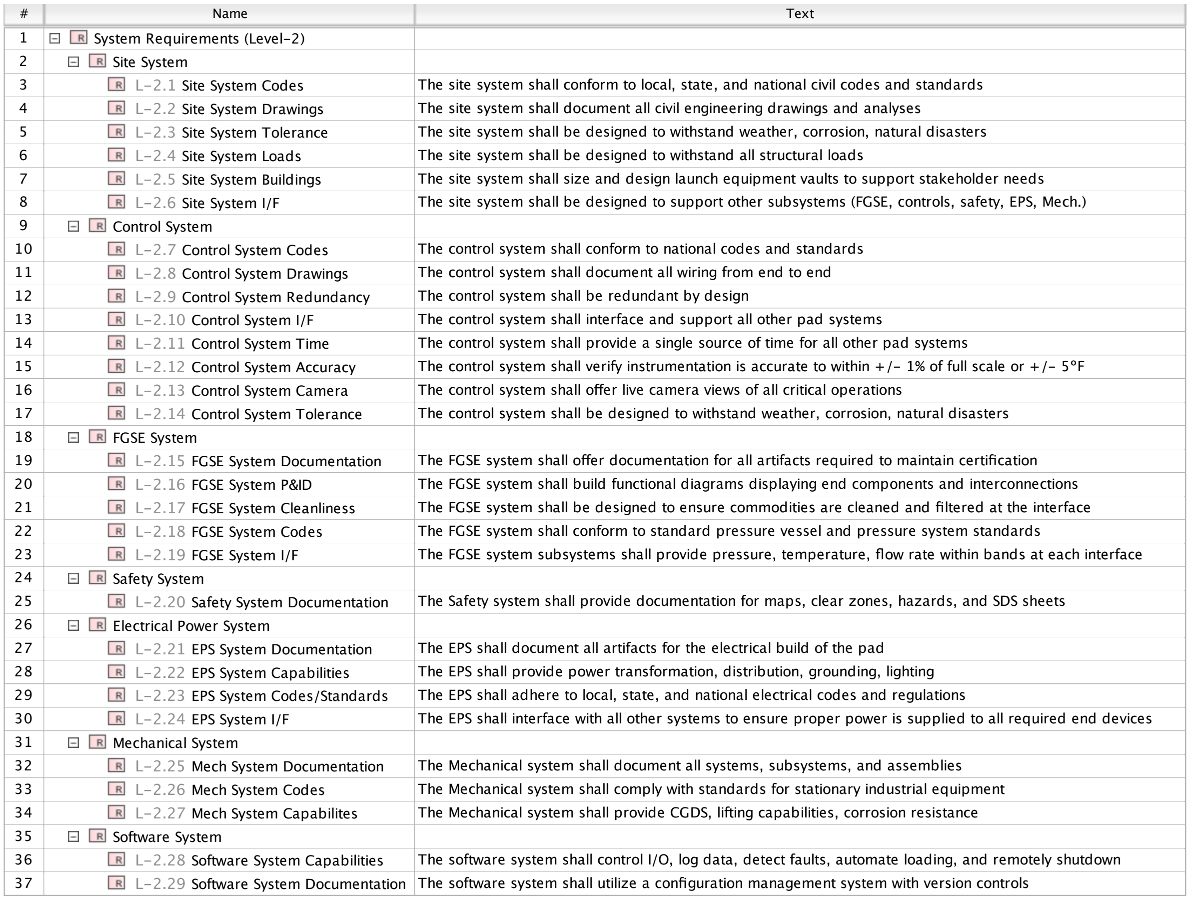
Legend
X Satisfy

Artifacts		1 – Contract Award	2 – System Requirement Review	3 – Preliminary Design Review	4 – Critical Design Review	5 – Delta CDR	6 – Test Readiness Review	7 – Ground Breaking	8 – Construction Complete	9 – Mechanical Checkout	10 – System Certification	11 – Electrical Checkout	12 – Controls Channelization	13 – System Level Leak Checks	14 – Performance Testing	15 – Wet Dress Rehearsal	16 – Launch
		1	2	5	8	8	3	3	2	2	1	1	1	1			
R 1 1-Mission Requirements	3	X	X	X													
R 2 2-Mission Analysis	3		X	X	X												
D 3 3-System Specification	3			X	X	X											
D 4 4-System Architecture	3			X	X	X											
D 5 5-Subsystem Architecture	2				X	X											
R 6 6-Component Specification	3				X	X	X										
R 7 7-Component Design	2				X	X											
R 8 8-Flow Analysis	3			X	X	X											
R 9 9-Engineering Drawing Review	4				X	X	X	X									
R 10 10-ICD Review	5					X	X	X	X	X							
R 11 11-System Test Plan	7							X	X	X	X	X	X	X			

Cells are marked with an ‘X’ to identify target completion dates for the artifact. For example, when the contract is awarded, attention is placed on developing the requirements and system specifications. As the project matures, the requirements become more solidified and the engineers shift their attention towards the architecture, component specifications, design, and system test plans in preparation for the critical and final design reviews. When the artifacts versus milestones chart is completed, it is clear that the bulk of the design work, which is the predecessor to the physical build, must be done in a short amount of time to support successors. Throughout the life cycle of the project, cost, quality, and schedule are constantly being weighed against one another.

The software modeling method is MBSE using SysML and the tool used for this project is Cameo Systems Modeler. For the introductory examples in Chapter 2, the open-source industrial-grade software by Eclipse titled Papyrus was investigated. Other popular SysML/MBSE modeling tools include IBM Rhapsody, Capella Open Source, Enterprise Architect, Innoslate, CORE, Modelio, and SysML Designer [50]. The modeling effort requires an organized approach with a clear configuration management strategy implemented. Ideally, multiple systems engineers will have the ability to check out the project and work simultaneously on the same living document. Training the modeling team is also vital to mission success, and the manager of the project must allow time for the SMEs to become well-versed and comfortable with the model-based software. During the early stages of the project, a subset of the engineering team will perform the majority of the modeling effort. This core group is responsible for maintaining the overall integrity of the model through different life cycle stages.

Once mission level requirements are defined, the next step is to analyze the requirements by reviewing, assessing, balancing, and prioritizing stakeholder requirements and transforming them into a technical or functional view showing how the stakeholder needs may be met. The strategy chosen to accomplish this is by defining Level-2 requirements for each system and defining Level-3 requirements for each subsystem, captured in MBSE requirements tables. These system and subsystem objectives are linked to at least one mission (Level-1) requirement to ensure traceability. After the requirements have all been defined, the remainder of this section seeks to ascertain external elements of the mission that interact with each system being designed, define measures of effectiveness (MoE), and introduce the concept of 'Blackbox' requirements for future design iterations.



#	Name	Text
1	System Requirements (Level-2)	
2	Site System	
3	L-2.1 Site System Codes	The site system shall conform to local, state, and national civil codes and standards
4	L-2.2 Site System Drawings	The site system shall document all civil engineering drawings and analyses
5	L-2.3 Site System Tolerance	The site system shall be designed to withstand weather, corrosion, natural disasters
6	L-2.4 Site System Loads	The site system shall be designed to withstand all structural loads
7	L-2.5 Site System Buildings	The site system shall size and design launch equipment vaults to support stakeholder needs
8	L-2.6 Site System I/F	The site system shall be designed to support other subsystems (FGSE, controls, safety, EPS, Mech.)
9	Control System	
10	L-2.7 Control System Codes	The control system shall conform to national codes and standards
11	L-2.8 Control System Drawings	The control system shall document all wiring from end to end
12	L-2.9 Control System Redundancy	The control system shall be redundant by design
13	L-2.10 Control System I/F	The control system shall interface and support all other pad systems
14	L-2.11 Control System Time	The control system shall provide a single source of time for all other pad systems
15	L-2.12 Control System Accuracy	The control system shall verify instrumentation is accurate to within +/- 1% of full scale or +/- 5°F
16	L-2.13 Control System Camera	The control system shall offer live camera views of all critical operations
17	L-2.14 Control System Tolerance	The control system shall be designed to withstand weather, corrosion, natural disasters
18	FGSE System	
19	L-2.15 FGSE System Documentation	The FGSE system shall offer documentation for all artifacts required to maintain certification
20	L-2.16 FGSE System P&ID	The FGSE system shall build functional diagrams displaying end components and interconnections
21	L-2.17 FGSE System Cleanliness	The FGSE system shall be designed to ensure commodities are cleaned and filtered at the interface
22	L-2.18 FGSE System Codes	The FGSE system shall conform to standard pressure vessel and pressure system standards
23	L-2.19 FGSE System I/F	The FGSE system subsystems shall provide pressure, temperature, flow rate within bands at each interface
24	Safety System	
25	L-2.20 Safety System Documentation	The Safety system shall provide documentation for maps, clear zones, hazards, and SDS sheets
26	Electrical Power System	
27	L-2.21 EPS System Documentation	The EPS shall document all artifacts for the electrical build of the pad
28	L-2.22 EPS System Capabilities	The EPS shall provide power transformation, distribution, grounding, lighting
29	L-2.23 EPS System Codes/Standards	The EPS shall adhere to local, state, and national electrical codes and regulations
30	L-2.24 EPS System I/F	The EPS shall interface with all other systems to ensure proper power is supplied to all required end devices
31	Mechanical System	
32	L-2.25 Mech System Documentation	The Mechanical system shall document all systems, subsystems, and assemblies
33	L-2.26 Mech System Codes	The Mechanical system shall comply with standards for stationary industrial equipment
34	L-2.27 Mech System Capabilites	The Mechanical system shall provide CGDS, lifting capabilities, corrosion resistance
35	Software System	
36	L-2.28 Software System Capabilities	The software system shall control I/O, log data, detect faults, automate loading, and remotely shutdown
37	L-2.29 Software System Documentation	The software system shall utilize a configuration management system with version controls

Figure 18: System Level-2 Requirements

The launch pad's Level-2 objectives reflect the values of the stakeholders with traceability to each system. The Level-3 objectives reflect the values of the stakeholders at a more detailed level with traceability to each subsystem. In the requirements tables, the systems at Level-2 include site, controls, FGSE, safety, electrical power, mechanical, and software. In the first iteration of Level-3, requirements are simplified by creating three subsystems which include codes and standards, documentation, and capabilities for each system.

As an example of traceability to the mission, a Level-1 objective is to deliver all consumables within the allowable ranges expected by the vehicle at the interface (L-1.15), and this is traced to the FGSE (L-2.19) and control systems (L-2.10) in Level-2. The site and mechanical systems must support the loads the spaceport will experience (L-1.8), with derived requirements for site (L-2.4) and mechanical (L-2.27) systems. The controls and software systems Level-3 derived requirements (L-3.21 and L-3.62) are necessary in order to fulfill the mission Level-1 requirement (L-1.11) of remote-control capability. The safety and environmental disciplines ensure that the spaceport will not put the personnel in a dangerous atmosphere where one could be fatally injured (L-1.7), and in order to accomplish this the safety group must document all hazard maps, clear zones, and safety data sheets within derived system level requirement (L-2.20). During the creation of Level-2 requirements, specific systems begin to trace to Level-1 requirements, allowing the team to gain a better understanding of the roles and responsibilities for each discipline. Figure 18 outlines a proposed first iteration of the system (Level-2) requirements. Level-3 requirements provide further decomposition. If a system or subsystem level requirement cannot be linked to a mission level requirement, then it should not exist. All requirements must have traceability to an overarching goal in order to justify implementation. After design iteration, the requirements tables will grow more complex.

#	Name	Text
1	Subsystem Requirements (Level-3)	
2	Site Subsystem - Documentation	
3	L-3.1 Site Engineering Dwgs	Site shall generate engineering drawings of for foundation, erosion & sediment control, and supports for other systems
4	L-3.2 Site 3D Dwg	Site documentation subsystem shall include a 3-Dimensional CAD model of the launch pad
5	Site Subsystem - Codes/Standards	
6	L-3.3 Site Code	Pad shall adhere to civil codes and standards GFSC-STD-8009, NASA-STD-8719.12A, and 49 CFR 177.848.
7	L-3.4 Building Code	Building design shall follow the International Building Code
8	Site Subsystem - Capabilities	
9	L-3.5 Vaults	Launch pad shall contain four launch equipment vaults
10	L-3.7 Plume Protect	Facility shall be built to protect all systems from rocket plume
11	L-3.8 Piping Support	Design to support cross country piping for all systems
12	L-3.9 EPS Support	The launch facility shall contain concrete pads for electrical power systems.
13	L-3.10 Cryogenic Vent	Pad shall contain a location to safely vent cryogenic liquids
14	L-3.11 Flame Trench	The launch facility shall contain a flame trench to direct the rocket plume away from the facility.
15	L-3.12 Flood	Facility shall be built in a 500-year flood zone, 6% chance of flood in first 30 years of operation
16	L-3.13 Thrust	Build the launch pad to withstand maximum thrust of rocket
17	Controls Subsystem - Documentation	
18	L-3.14 Single Line Drawing	The control documentation shall include a single-line-drawing for each piece of instrumentation
19	Controls Subsystem - Codes/Standards	
20	L-3.15 Time Protocol	The ground control network shall offer NTP (National Time Protocol)
21	L-3.16 Symbol Standard	The control documentation shall follow ANSI/ISA-5.1-2009 - Instrumentation Symbols and Identification
22	L-3.17 Calibration Cycle	The launch facility shall calibrate all critical I/O per the frequency recommended by each manufacturer
23	Controls Subsystem - Capabilities	
24	L-3.18 Server Racks	Each vault (total of 4) will contain quantity (2) server racks (8 racks total)
25	L-3.19 UPS Runtime	UPS backup power must last 30 minutes accounting for the apparent power of facility
26	L-3.20 Box Protection	All control and junction boxes that are outside will have a NEMA 4X rating or greater
27	L-3.21 Controller Amount	The launch facility will have a primary and redundant controller (PLC)
28	L-3.22 Cameras	The launch facility shall provide 4 cameras with PTZ >= 20X optical zoom
29	L-3.23 Controller Location	The pad controllers and junction boxes shall be located inside the vaults
30	L-3.24 Controller Connect	Pad shall provide primary and redundant fiber and ethernet between all controllers
31	FGSE Subsystem - Documentation	
32	L-3.25 FGSE Leak Tests	FGSE shall generate documentation for initial leak checks of all subsystems
33	L-3.26 FGSE Mechanical Checkout	FGSE shall generate documentation for all mechanical checkouts of all subsystems
34	L-3.27 FGSE Isometrics	FGSE shall generate isometrics detailing dimensions of all components, sub-assemblies, and assemblies
35	L-3.28 FGSE Inspection Plan	FGSE shall generate documentation detailing out the inspection plans for each commodity
36	L-3.29 FGSE Relief Valve Analysis	FGSE shall perform and document relief valve analysis for all subsystems
37	L-3.30 FGSE Fabrication	FGSE shall document materials of construction, material test reports, heat number, and certifications of conformance
38	L-3.31 FGSE Weld Procedures	FGSE shall generate weld procedures, qualified to B31.3 for the pressure, temperature, and material of construction
39	L-3.32 FGSE Non-Destructive Exam	FGSE shall document all non-destructive testing and examinations of certified components
40	L-3.33 FGSE Hydrostatic Testing	FGSE shall document all hydrostatic testing reports required for system certification
41	L-3.34 FGSE P&ID	FGSE shall generate and maintain Piping and Instrumentation (P&ID) drawings for all commodities
42	FGSE Subsystem - Codes/Standards	
43	L-3.35 Cleanliness	All components shall meet SAE AS4059 Rev E. Class 2 cleanliness standard
44	L-3.36 PVS Standard	FGSE must follow standard NASA STD 8719.17 NASA Requirements for ground based pressure vessels and pressurized systems
45	L-3.37 Sample Frequency	All commodities interfacing with launch vehicle shall be sampled 1 month prior to each mission
46	L-3.38 Pressure Vessel Code	Pressure vessels shall conform to ASME B&PV Section VIII Division 1 standard
47	FGSE Subsystem - Capabilities	
48	L-3.39 Commodity Fill Skids	All commodity storage areas shall have a fill skid capable of refilling commodities
49	L-3.40 Fuel Specs	Storage=30k gallon, flow=0-100 gpm, MEOP=150 psig, filter=10 µm, GN2 purge
50	L-3.41 HVAC Specs	Compressed Air, flow=0-100 scfm, MEOP=200 psig, filter=10 µm, Temp 0-100°F
51	L-3.42 Water Specs	Storage=100k gallon, flow=0-100 gpm, MEOP=150 psig, Flow Duration=60 seconds
52	L-3.43 Helium Specs	Storage=100k scf, flow=0-15 scfm, MEOP=5k psig, filter=10 µm
53	L-3.44 Oxidizer Specs	Storage=60k gallon, flow=0-50 gpm, MEOP=150 psig, filter=10 µm, GHe purge, Temp<=-297°F
54	L-3.45 Nitrogen Specs #1	Storage=100k scf, flow=0-100 scfm, MEOP=5k psig, filter=10 µm
55	L-3.46 Nitrogen Specs #2	Nitrogen shall be supplied to electro-pneumatic control valves between 70-120 psig for actuation.
56	Safety Subsystem - Documentation	
57	L-3.47 Hazard Zones	Safety shall develop a hazard map classifying hazardous areas in a 3-dimensional format
58	L-3.48 Hazardous Ares	Safety shall identify all hazardous work areas and have protocols to mitigate the risks
59	L-3.49 Fire Extinguish	The launch pad shall conform to NFPA 10 Standards for Portable Fire Extinguishers
60	EPS Subsystem - Codes/Standards	
61	L-3.50 EPS Code	The launch pad shall conform to NFPA 70 National Electric Code
62	EPS Subsystem - Capabilities	
63	L-3.51 MDP	EPS shall contain a main distribution panel sized to safely operate all facility hardware
64	L-3.52 Power Spec	EPS shall accept 480 VAC and transform to 120 VAC, 208VAC, and 240VAC.
65	L-3.53 Lighting Spec	EPS shall contain lighting properly sized to safely work twenty-four hours a day
66	L-3.54 Grounding	EPS shall contain grounding points at each building and at all skids
67	EPS Subsystem - Documentation	
68	L-3.55 EPS Documentation	EPS shall document single line drawings, panel schedules, feeder schedules, grounding points, lighting, power distribution
69	Mech Subsystem - Documentation	
70	L-3.56 Mech Docs	Mechanical system shall provide artifacts including hydraulic circuit diagram, bill of materials
71	Mech Subsystem - Codes/Standards	
72	L-3.57 Hardware Spec	Pad shall build all structures and mechanisms with a 1.5 safety factor or greater.
73	L-3.58 Mech Code	The hydraulic system shall comply with ANSI Hydraulic Systems Standard for Stationary Industrial Machinery NFPA/JIC T2.24.1-1990
74	Mech Subsystem - Capabilities	
75	L-3.59 Corrosion	Pad shall run a semi-annual corrosion control plan to protect all pad hardware from degredation.
76	L-3.60 CGDS	Pad shall contain a combustible gas detection system capable of alerting pad personnel of a leak.
77	L-3.61 Lift Spec	Pad shall offer a hydraulic lift circuit to lift and lower rocket with a mass up to 300,000 lb
78	Software Subsystem - Capabilities	
79	L-3.62 Remote I/O	Software HMI shall report up to 500 I/O channels at an update rate of 10 Hz simultaneously
80	L-3.63 Data Logging	Software shall have the ability to log all launch pad I/O channels 24/7 at a rate of 1 Hz
81	L-3.64 Fault Detect	Software shall detect faults utilizing error codes, loop-back testing, and notify operator on the HMI
82	L-3.65 Automate Flows	Software shall automate control loops to remotely fill and detank all commodities
83	L-3.66 Remote Shutdown	Software shall be capable of remotely safing the launch pad
84	Software Subsystem - Documentation	
85	L-3.67 Software Config	Software shall utilize configuration management with three branches (master, develop, hotfix)

Figure 19: Subsystem Level-3 Requirements

During each mission, the launch pad is expected to interact with a unique set of external elements, internal elements, systems, subsystems, and stakeholders to fulfill mission, system, and subsystem requirements. This is referred to as the mission context. A mission context typically varies from mission to mission. For example, the launch pad may work with different payload companies for two separate rocket launches where each user requires unique payload interface conditions. The internal elements identified above as system or subsystem objectives within the requirements tables may require tailoring per mission. The external elements for the first mission will include the rocket company (LSP), NASA, the payload customer, the fire department, ground-based radar stations, the integration facility, the general public, the Earth (e.g. weather), control operators, and external networks or communication links. Each of these external elements may be modeled as blocks within a BDD with the launch pad being the system of interest tied to all of the external blocks. In order to fulfill all Level-1 (mission), Level-2 (system), and Level-3 (sub-system) requirements for a particular mission, it is imperative to have a clear understanding of both external and internal blocks. Between missions, the requirements tables must be reviewed, and deltas will be captured with improved traceability using MBSE.

MoE and Blackbox requirements are two tools which help engineers to analyze and optimize mission, system, and subsystem requirements. An MoE is a metric designed to correspond to the achievement of a desired result or the accomplishment of a mission-level objective. The metric helps the team evaluate aspects of a mission such as behavior, capability, achievement [18]. For the spaceport, preliminary MoEs for the first iteration could include code compliance, safety statistics, launch cadence, ambient temperature during launch, interface conditions, structural state, and end result of the mission. An MoE may be decomposed into Measures of Suitability (MoS) and Measures of Performance (MoP) [28].

Black box requirements help to specify performance, physical, interface, and functional requirements observable and verifiable at a mission level [18]. The interface conditions or the runtime of the EPS are examples of Level-1 requirements that may be measured while observing the launch pad as a black box. The requirement to output and log data is observable as an output of the launch pad. In comparison, the requirement for the control system to process information from an end device to produce a derived measurement which is used to determine pad state is not a black box requirement. With the proper level of abstraction, the launch pad design is not over-constrained, and this enables the designers to explore alternative methods to achieve end goals. In turn, this helps to drive a more cost-effective and efficient solution. Due to this, it is helpful to build a list of the functions which may be realized by alternative designs. Only the critical quality, performance, and physical characteristics are shown in the black box specifications. The keyword ‘Technical Performance Measures’ (TPM) represents system properties which have the ability to heavily impact a mission as well as system performance. While MoE is applied to mission-level performance parameters, the term TPM is used with respect to system level performance parameters [18]. With the launch pad being a system of interest represented as a block, the following black box measures are specified in Table-7.

Table 7: Launch Pad Black Box Specification Examples

Values	Operations	Ports	States
<<tpm>> FGSE I/F Conditions	Run Auto-sequences	EPS I/F	Standby
<<tpm>> EPS I/F Conditions	Derive Measurements	Controller I/F	Pressurization
<<tpm>> FGSE Storage	Load commodities	Range I/F	Control Flow
<<tpm>> Cost	De-tank	Vehicle I/F	Testing
<<tpm>> Data	Sample FGSE I/F	IT I/F	Securing
<<tpm>> Reliability	Video Monitoring	Video I/F	Safe State

3.2 Architectural Design Process

The next step in the ISO/IEC 15288 technical process flow is to design a logical architecture using MBSE with the purpose of synthesizing a solution which fulfills all stakeholder requirements. Inputs to this step will include functional and performance requirements, architectural constraints, a traceability matrix, and system interface specifications. Formal outputs of this process may include an architectural design baseline, system element descriptions, interface requirements, a refined traceability matrix, and a strategy for verification [49]. In order to produce these outputs, sub-steps completed in the first design iteration will include setting up the model, establishing a reliable design convention, organizing the model via the appropriate package structure, performing an initial mission analysis, and building IBDs of systems and subsystems in SysML. Defining a logical architecture facilitates navigation, reuse, and access control. The system model for the launch pad is organized into the package structure outlined in Figure 20. The model's package structure and organization is simplified to illustrate the concept. As the design progresses, a more robust model organization method is an expected output of the exercise.

The launch pad model consists of a top level package with nested containers for individual elements and artifacts. The elements in separate packages may be easily related and traced to one another through an assortment of relationship types. Examples of relations include abstraction, allocation, dependency, derived requirement, refine, satisfy, and trace. For a full list of the relationship types and their intended application, the modeling team should refer to the technical user manual for the software selected to ensure proper implementation. The top level package in the proposed MBSE architecture is titled the 'Launch Pad Mission Context', with sub-packages defining the mission-level directories such as Analysis, Behavior, Use Cases, Structure, and a Black Box Specification. The Supporting Elements package includes Interface Definitions, Value Types, and Viewpoints. ‘Interface Definitions’ holds model elements such as port definitions, signals, and flows, intended to be reused acrossed disciplines [18]. ‘Value Types’ is a library of units and quantities standardized acrossed the model. Viewpoints pertains to accurately identifying stakeholders’ interests. The packages are structured based on the mission, system, and subsystem requirements of Levels-1, 2, and 3. Not all packages are shown, and there are many subsystems and elements not represented.

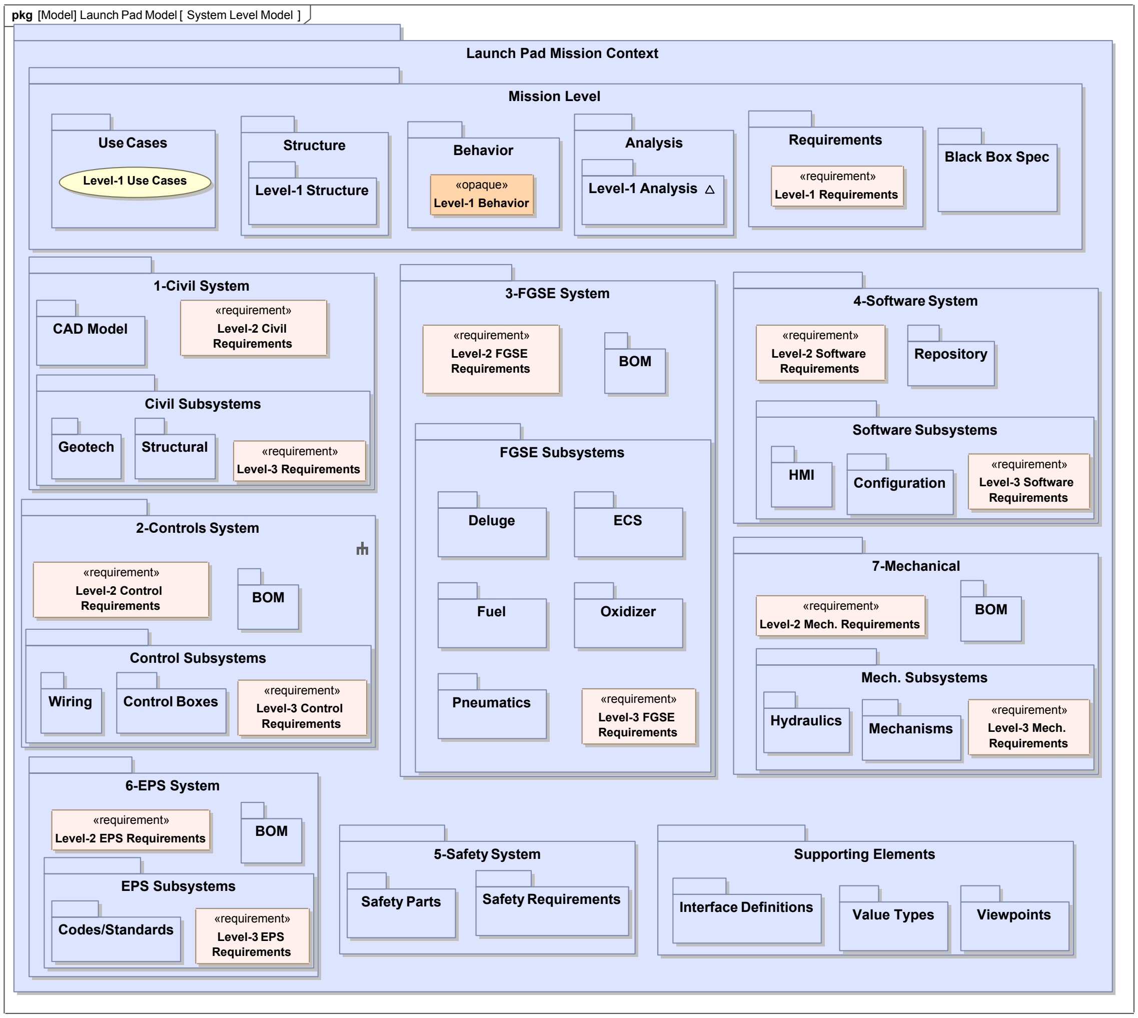


Figure 20: Launch Pad Model - Package Structure

Setting up the model involves the establishment of a modeling convention. Conventions for naming, diagram layout, annotation, interface designations, and proper selection of diagram types depending on the application are critical to establish early in the technical process to reduce confusion in the MBSE environment. For this launch pad model exercise, Table-8 gives examples of proposed conventions [18].

Table 8: Launch Pad Modeling Conventions [18]

Launch Pad Modeling Conventions	
#1	Activities, Blocks, and Classification Elements start with upper case (e.g. Regulator)
#2	Port names begin with i/f representing an interface (e.g. i/f +24VDC power rail)
#3	Activity names shall be defined using a verb followed by a noun (e.g. Open-Valve)
#4	Actions, parts, and properties of elements shall start with lower case (e.g. channel# 18)

Once the model is set up, conventions are established, and packages are created, the first SysML diagrams to create shall be at the mission level. As an example, a mission level event timeline is outlined in Figure 21 beginning approximately one month prior to the rocket arriving at the launch pad. During this time, commodities are topped off, samples at interfaces with the vehicle are taken to verify cleanliness, fuel is conditioned, electrical tests are performed to verify pad health, and requirements begin to close. Once the rocket arrives, it is mechanically mated to the launch platform, and the HVAC system begins a 24/7 constant purge of vehicle cavities to ensure positive pressure, temperature, humidity, and cleanliness. An activity diagram is utilized to portray the flow of events. The activities within this diagram are considered the top-level mission functions required in order to satisfactorily accomplish the mission. By capturing mission level diagrams during the initial definition of the logical architecture, this promotes mission level awareness and subject matter expert awareness of the overarching goals.

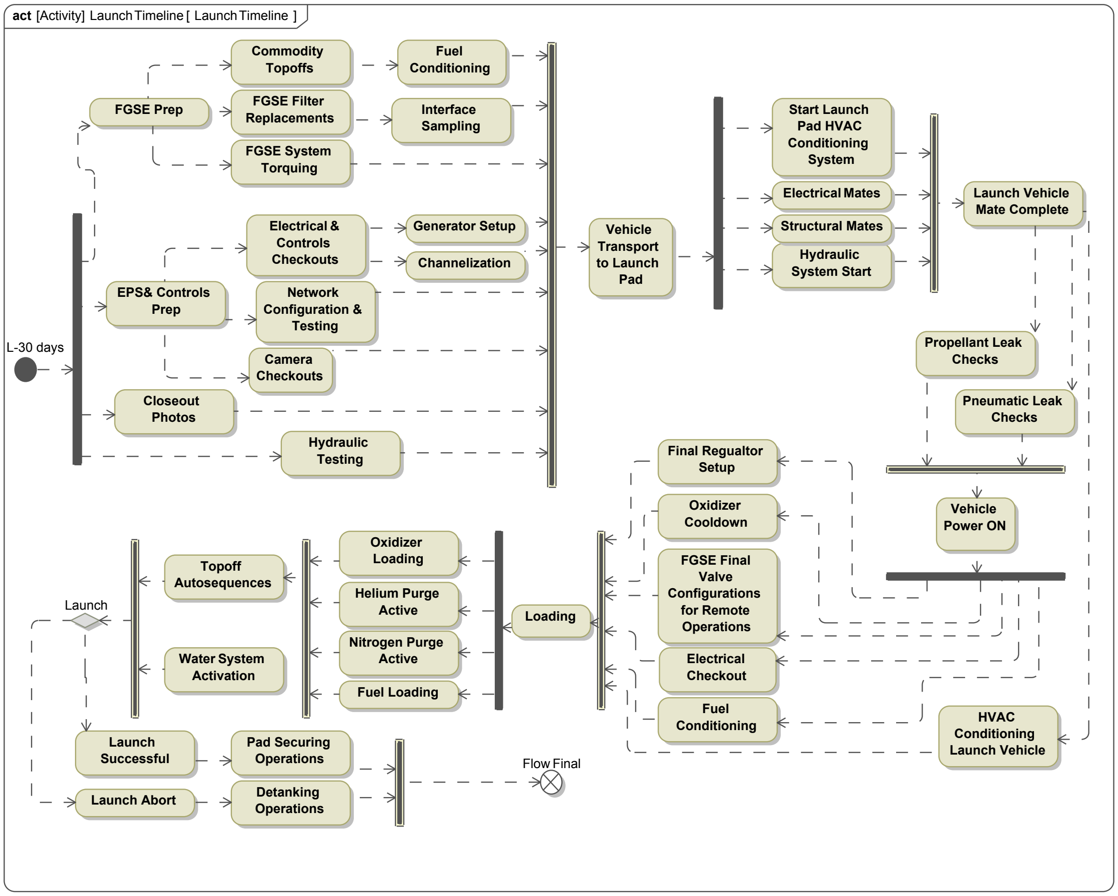


Figure 21: Activity Diagram of Proposed Launch Timeline (L-30 to T-0)

It is also important to understand early in the model creation that while the LSP and the launch pad share common goals, the overall mission for the launch facility is different compared to the mission for the rocket. This model is tailored to focus on ground systems, not the launch vehicle. From the launch pad's perspective as a mission stakeholder, a successful mission is considered complete once the vehicle takes off, interface conditions are verified to be within the allowable ranges, and all of the Level-1 requirements are verified to be successful. The top failure modes that have the highest probability of causing a mission failure are imperative to understand as early as possible in order to reduce the probability of failure. The failures should attempt to be mitigated and weighted based on risk level and likelihood of occurrence. An example of a failure mode is the loss of a flow control valve on the propellant system that is needed to fill onboard tanks. In order to meet this Level-3 requirement, the launch pad must provide a flow rate between 0-100 gallons per minute. If the spaceport is designed without redundancy on the flow control system, the control valve which maintains the flow rate is considered a single point failure. By understanding this potential failure mode early in the formation of the project's architecture, the subject matter expert designing the fuel system may take this observation and design for redundancy. A mission level failure chart depicted via a BDD within Figure 22 recognizes a small subset of the potential mission failure modes.

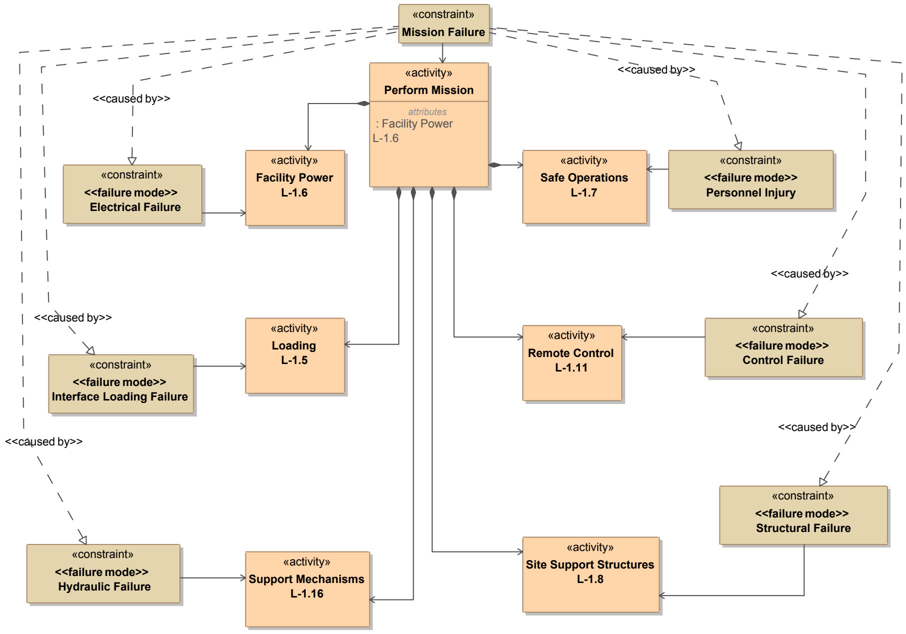


Figure 22: Potential Launch Pad Failure Modes

Once mission level SysML diagrams have been created and a package structure for each system has been defined, the next step is to decompose the launch pad systems into individual parts within BDDs, identify system components, define how systems interface with one another, and display part interconnection with the use of internal block diagrams. The first iteration of the system decomposition into subsystems and components will be performed for a subset of the systems identified. Connections link systems and subsystems to one another to represent interfaces and shared functions. In SysML diagram examples, connectors between software, controls, FGSE, and other subsystems may not be shown in order to simplify the instance of the block definition diagram. The model has the ability to show or hide connections without deleting the underlying associations.

In Figure 23, the internal block diagram details the interconnections between systems. Interfaces between the general launch pad, the launch vehicle, and systems are connected with ports. Examples include the grounding interface between the electrical and civil subsystems, the hydraulic controls interface between the mechanical and control subsystems, and the FGSE interfaces for multiple commodities between the control subsystem and the launch vehicle (LV). This is the first iteration of an interconnection diagram between subsystems. As design iterations continue, subsystem interfaces may change. An advantage to a properly configured MBSE model is the capability for the modeling environment to automatically reflect these design changes. Ports interconnecting the spaceport to the launch vehicle include external electrical and mechanical interfaces. Prior to launch when the rocket is on the pad, telemetry and commands travel to and from the spacecraft via an electrical interface internal to the launch vehicle. The commands and telemetry continue propagation from launch vehicle to the spaceport through an electrical umbilical denoted as the ‘LV electrical i/f’. The physical hardware representing this interface is typically a custom-built electrical harness which complies with the safety system's hazard classification. During LV ascent, the spacecraft will resume communication with the launch facility through the use of vehicle RF communication, and the data is further relayed to the ground station. After spacecraft separation from the LV, an antenna is deployed, and communication resumes through alternate ground command and data ports. From the system level interconnection diagram, it is clearly shown that there are numerous interfaces between the launch pad, the rocket, the payload, and other external entities which must work together to satisfy Level-1 requirements.

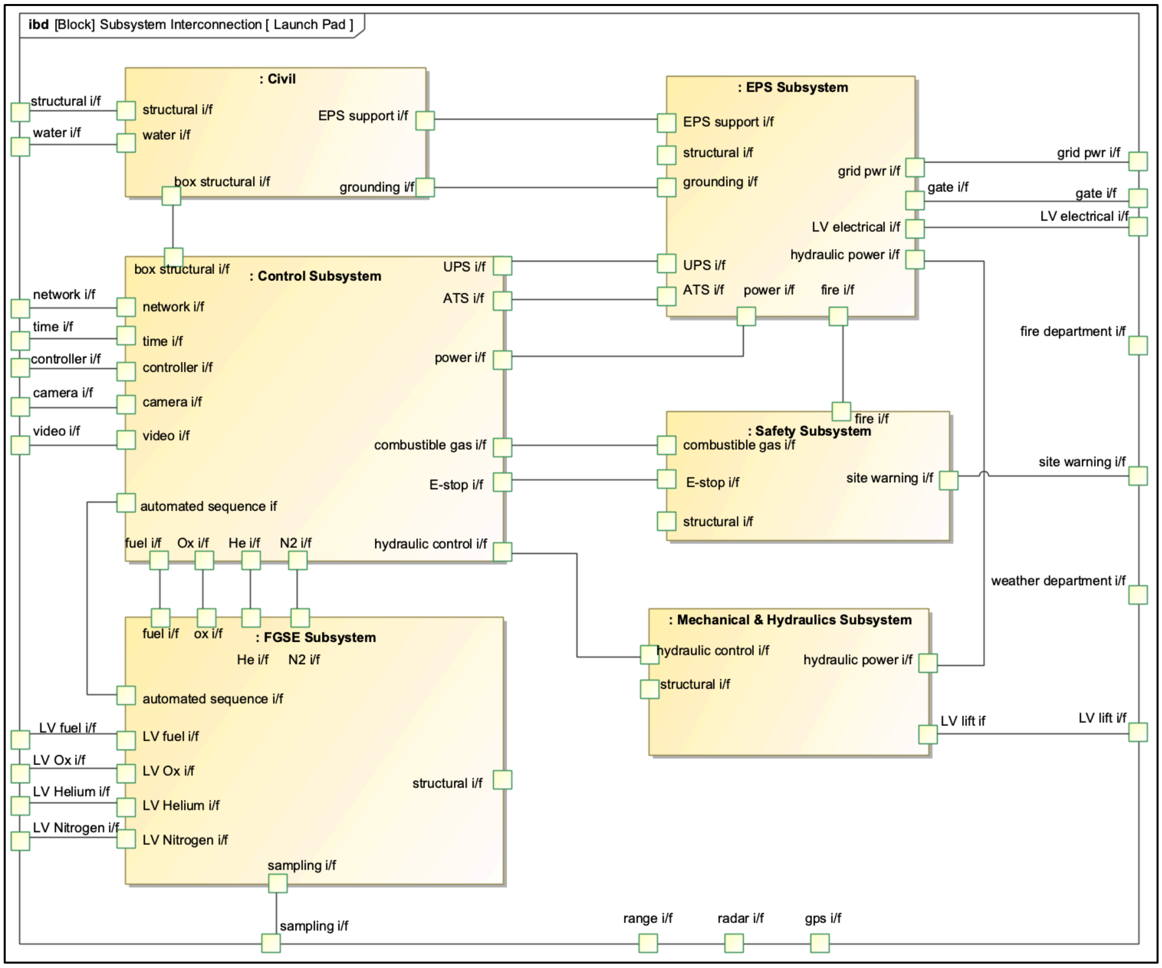


Figure 23: Launch Pad Subsystem Interfaces

3.2.1 Civil System Architectural Design

The civil system creates, improves, and protects the spaceport environment. Key functions include planning, design, and oversight of the construction and maintenance of the building structures and infrastructure on the launch pad such as roads, concrete foundations, structural supports, conduit, cable tray, miscellaneous metals, irrigation, and tank supports, and piping support. To visualize the spaceport and provide an architectural design to support all other subsystems, a 3D model is generated in a Computer Aided Design (CAD) program. For a commercial company to use a specific launch pad, details such as vehicle weight, fuel weight, thrust, launch mount weight, flame deflector weight, and any other large loads must be considered. It is the responsibility of the civil engineer to account for these loads and ensure safety factors are robust to ensure reliability and repeatability. The proposed launch pad will be constructed with four main sections (fuel storage, oxidizer storage, electrical storage, and launch mount).

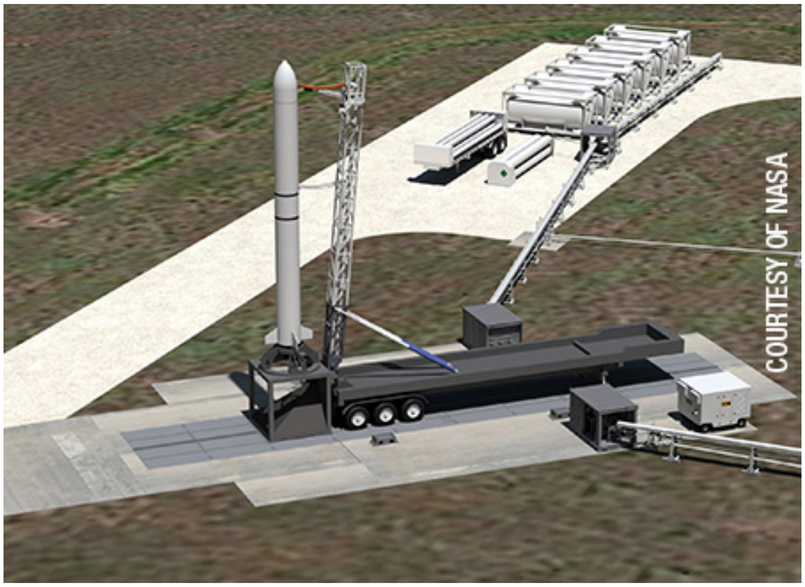

COURTESY OF NASA

Figure 24: NASA KSC 3D Model of Launch Pad 39C for Small Class Payloads [12]

The four sections will be made from 4,000 psi Portland cement that is thick enough to offer a weight capacity greater than 200,000 lb. to account for the maximum thrust profile of a small-class customer. Foundation design shall support reactions from the flame-deflector and structural loads experienced during hurricanes or tropical storms when a rocket is not present. The load types designed for shall include dead, live, blast, acoustic, thermal, and wind. In future iterations of the design, civil engineers shall attempt to incorporate the ability to perform static engine testing on the launch pad, which is often performed to certify launch vehicle systems prior to flight. Along with LV forces, the design team studies effects of a mission on the surrounding area. Examples include permitting, storm water design, and direction of exhaust dependent upon the surrounding area [8].

3.2.2 Control System Architectural Design

The control subsystem interconnection diagram is shown in Figure 25. The subsystem's purpose is to provide remote control of hazardous operations, automate operational processes, help operators establish situational awareness, offer hazardous or critical event detection, and to record and analyze data. The proposed design uses a primary and redundant server. The servers are synchronized, allowing the redundant computer to constantly record the state of the launch pad and pick up control if an anomaly is present on the primary. There is a fiber optic connection to an off-site location for remote control of the spaceport. The electrical vault is modeled as a block which contains the servers, network switches, fiber patch panels, and is the central information hub where all command and telemetry travels to and from. There are fiber optic connections to three controllers within the field located at the fuel storage, oxidizer storage, and launch mount areas. The three field controllers have multiple network connections, increasing redundancy by offering two independent paths back to the electrical vault. Each field controller accepts telemetry from the field for valve positions, pressure, temperature, flow, level, and other signal types by using analog inputs and digital inputs. Controllers also send commands to end devices using analog outputs and digital outputs. The digital inputs and outputs are 0-24 V DC. The analog inputs and outputs operate over 4-20mA current loops. With the control system IBD and fundamental voltage, current, and power specifications established, the responsible engineer has the ability to select hardware, generate control drawings, and implement a working system. From a safety perspective, the hardware selected shall be explosion-proof or intrinsically safe to comply with NFPA 70 Article 500.7 regarding hazardous location hardware compliance techniques.

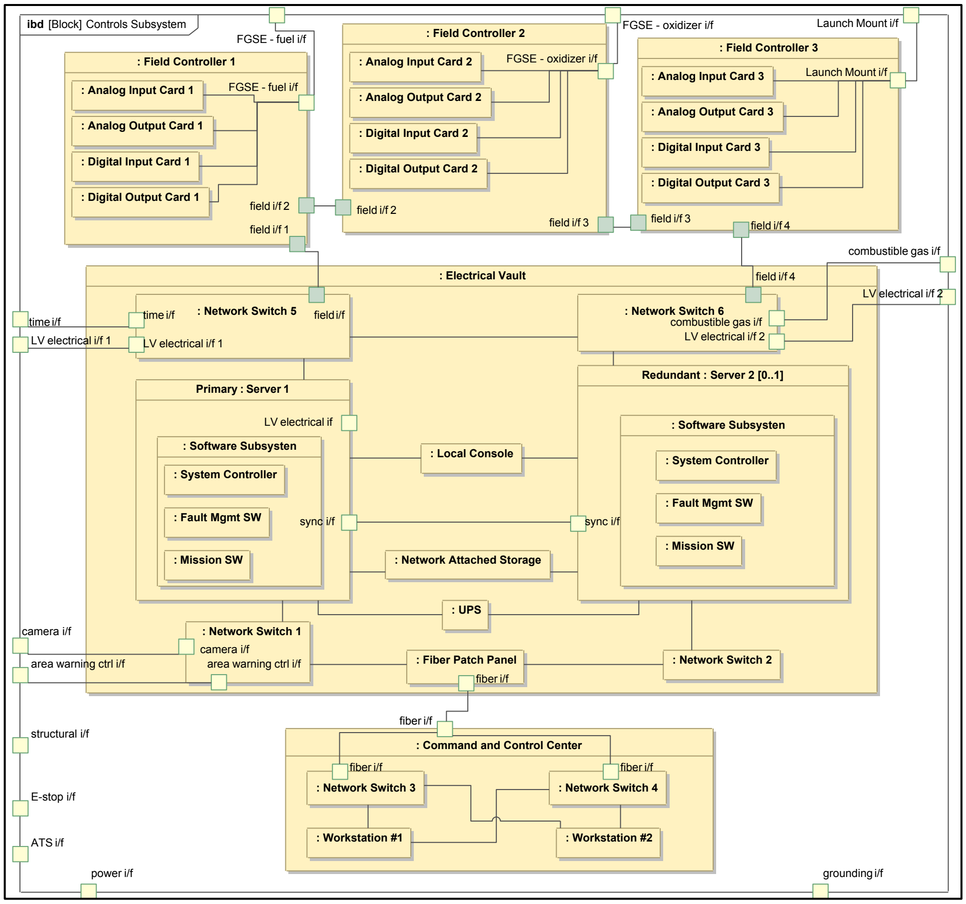


Figure 25: Controls IBD Interconnections and Blocks

3.2.3 FGSE System Architectural Design

The fluid ground support equipment stores, transports, filters, and loads liquid and gas commodities onto the LV. There are numerous FGSE subsystems and commodities to consider, dependent upon the specific LV being loaded. This launch pad is tailored to supply a small-class liquid LV with the selected fuel and oxidizer of RP1 (Kerosene) and LO_2 (LOX). The LV requires helium and nitrogen gas provisions to purge cavities, maintain cleanliness, reduce geysers within onboard oxidizer tanks, and actuate control valves onboard the rocket and on the ground systems. The spaceport requires a water system to suppress acoustics and reduce launch mount temperature. Typically rockets require additional commodities, but for this iteration only four subsystems (RP1, LOX, GN_2, GHe) are modeled within an IBD. There are several trade studies to be performed by the engineering team to meet a multitude of interface requirements.

The fundamental architecture for a subset of the FGSE system is represented by elements required for the fuel, oxidizer, nitrogen, and helium subsystems. Purge interfaces for fuel are represented by ‘GN2 i/f #’ notation and purge interfaces for oxidizer are represented by ‘GHe i/f #’. The IBD format progresses from left to right, starting with fill skids and storage areas, then continuing toward the launch vehicle interface points on the right side of each IBD. Multiplicity is annotated next to each hardware type to help users quickly understand how many of each component type are required on each skid. In future design iterations, more detail shall be added to each block such as unique identifiers and block attributes. Examples include sensor ranges, regulator flow characteristics, flow coefficients for valves and regulators, pressure setpoints for relief valves, maximum allowable operating pressure of individual components, tubing and piping dimensions, hazardous area mitigation techniques deployed, and tank specifications.

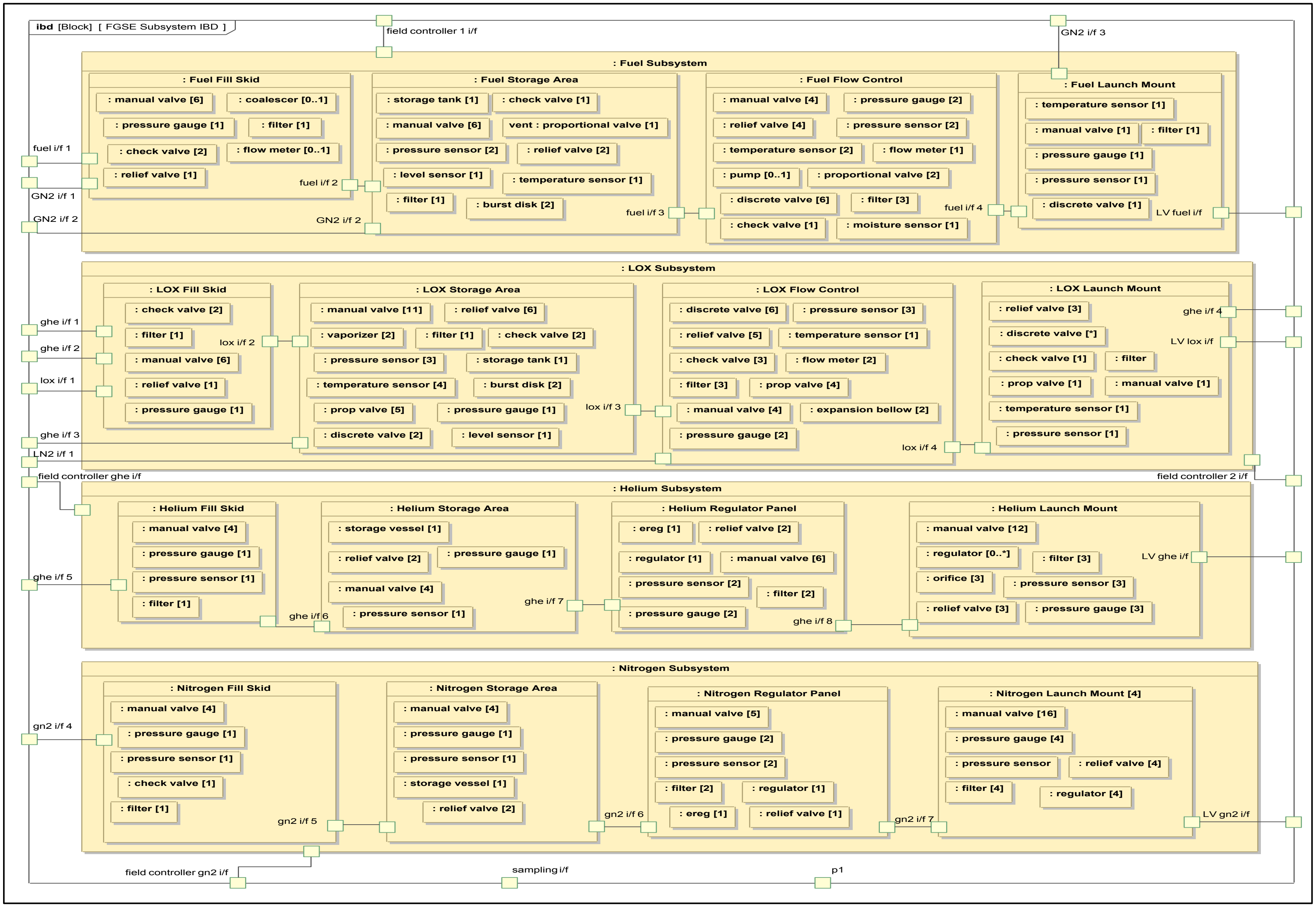


Figure 26: FGSE IBD Interconnections and Blocks

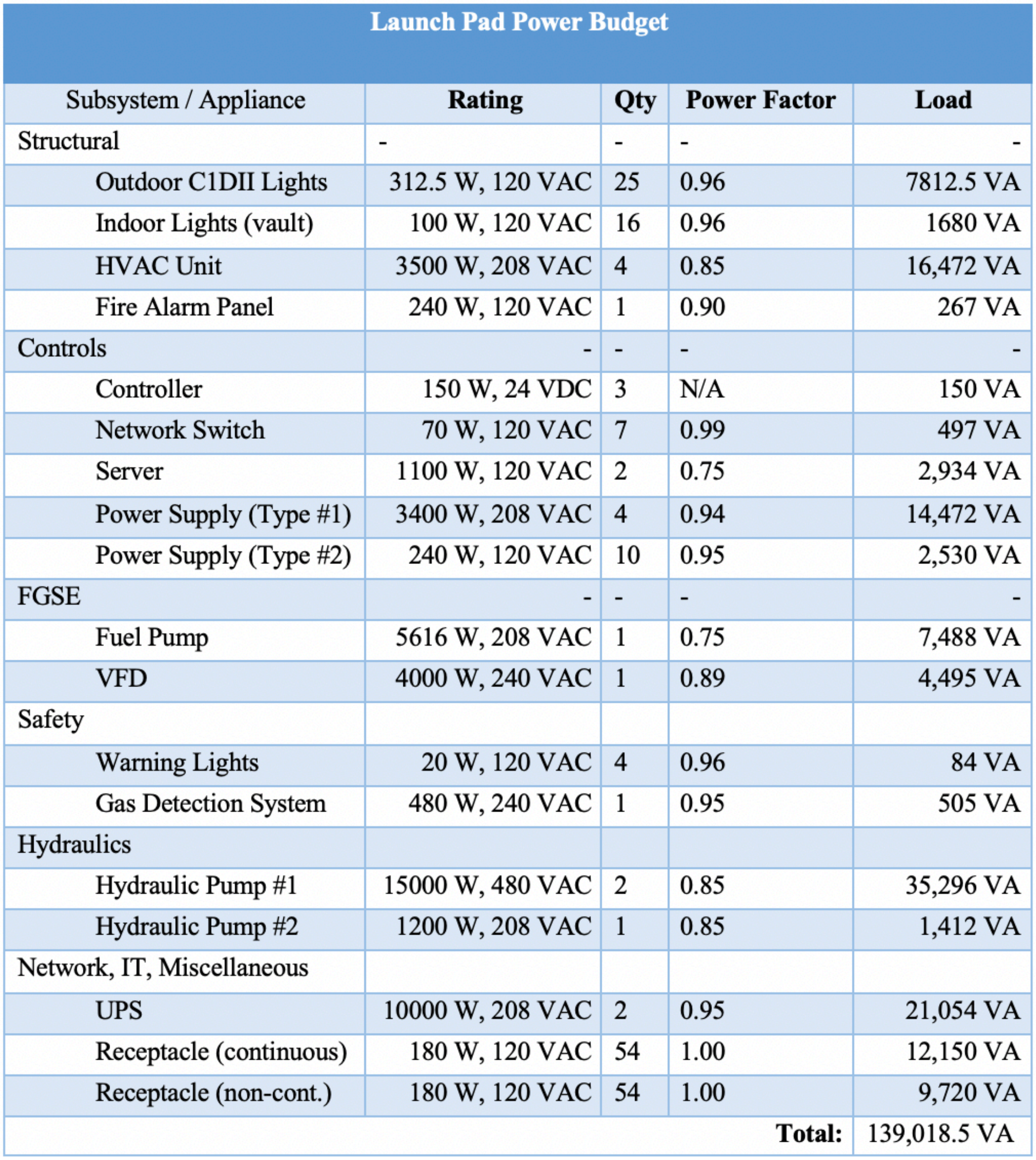
3.2.4 EPS Architectural Design

The electrical power subsystem is responsible for providing, storing, and distributing launch pad electrical power. The EPS also designs the lightning protection systems and grounding infrastructure. Architectural artifacts of the EPS include a site power plan, lighting pan, conduit layout, fire alarm plan, grounding drawing, lightning protection plan, single line drawing, panel schedule, alarm plan, and feeder schedule. In order to size the transformer and main distribution panel, the SME must identify the electrical power loads to support the mission. Electrical loads are typically generated by every other discipline, and these loads will vary depending on the operations taking place. The load characteristics should be determined early in the architectural design process to select the proper power distribution and voltage levels, sufficient space, and ventilation.

In addition to facility power, the spaceport typically incorporates redundant power sources such as UPS systems or generators depending on the budget, requirements, and reliability of the grid. The apparent power estimates of the field determine the size of the power source. An attribute of each block in the launch pad model that requires power should be the estimated load of the device. Once hardware is selected, the summation of this block characteristic is an example of increased efficiency for the electrical engineer using MBSE. As users modify the hardware selected, the model automatically recalculates the total load based on the summation of the load attributes within a parametric diagram. For the first iteration of the launch pad EPS, a power budget is represented within Table 9 to begin the design process. For the critical design review and later life cycle stages, all of the other EPS artifacts are required.

The power budget reveals that an estimated 139 kilovolt-amperes (kVA) of apparent power will be required to operate the launch pad. The proposed hardware operates at various voltage levels, including 24 VDC, 120 VAC, 208 VAC, 240 VAC, and 480 VAC. Transformers shall be selected to support end device needs. The power factor, or ratio of real power performing work to the apparent power supplied to the circuit, is factored into the calculation when applicable.

Table 9: Launch Pad Power Budget



Launch Pad Power Budget				
Subsystem / Appliance	**Rating**	**Qty**	**Power Factor**	**Load**
Structural	-	-	-	-
Outdoor C1DII Lights	312.5 W, 120 VAC	25	0.96	7812.5 VA
Indoor Lights (vault)	100 W, 120 VAC	16	0.96	1680 VA
HVAC Unit	3500 W, 208 VAC	4	0.85	16,472 VA
Fire Alarm Panel	240 W, 120 VAC	1	0.90	267 VA
Controls	-	-	-	-
Controller	150 W, 24 VDC	3	N/A	150 VA
Network Switch	70 W, 120 VAC	7	0.99	497 VA
Server	1100 W, 120 VAC	2	0.75	2,934 VA
Power Supply (Type #1)	3400 W, 208 VAC	4	0.94	14,472 VA
Power Supply (Type #2)	240 W, 120 VAC	10	0.95	2,530 VA
FGSE	-	-	-	-
Fuel Pump	5616 W, 208 VAC	1	0.75	7,488 VA
VFD	4000 W, 240 VAC	1	0.89	4,495 VA
Safety				
Warning Lights	20 W, 120 VAC	4	0.96	84 VA
Gas Detection System	480 W, 240 VAC	1	0.95	505 VA
Hydraulics				
Hydraulic Pump #1	15000 W, 480 VAC	2	0.85	35,296 VA
Hydraulic Pump #2	1200 W, 208 VAC	1	0.85	1,412 VA
Network, IT, Miscellaneous				
UPS	10000 W, 208 VAC	2	0.95	21,054 VA
Receptacle (continuous)	180 W, 120 VAC	54	1.00	12,150 VA
Receptacle (non-cont.)	180 W, 120 VAC	54	1.00	9,720 VA
			Total:	139,018.5 VA

3.2.5 Mechanical Architectural Design

The mechanical and hydraulic subsystem has the requirement to lift and lower the rocket where the maximum weight is up to 300,000 pounds (L-1.16, L-3.61). A hydraulic system architecture must be synthesized to accomplish this objective. A hydraulic cylinder, or linear actuator, gives a linear force output and produces movement by controlling fluid conditions. The force produced by a cylinder is equal to the pressure of the internal fluid multiplied by the area of the piston. The cylinder's piston and rod velocity are a function of how quickly the hydraulic fluid enters or exits the rod or cap end of the cylinder. Based on the lift and lowering force requirement, the proper bore size is selected. A cylinder with a bore larger than 4 inches should incorporate a 1.5x safety factor. Based on push and pull force estimation tables [6], a 12-inch cylinder bore (piston area of 113.10 in^2) operating at 2000 psig will generate a theoretical push stroke force of 226,200 pounds. Cylinders produce more force during the push stroke than the pull stroke due to the reduction in area on the "rod" end. A 12-inch cylinder operating at 2000 psig with a 5.50-inch piston rod diameter (net rod end area of 89.342 in^2) generates a theoretical pull stroke force of 178,684 pounds. With a 1.5x safety factor and two cylinders of equal dimensions, the resulting 300,000-pound lift requirement is capable of being met in both the push and pull directions. It is best practice to reduce the number of cylinders in order to improve control of the system, resulting in two cylinders with larger diameters being selected for the primary iteration of the hydraulic hardware design study. It is important for the piston rod column strength to be suitable for the intended application. In order for the launch mount to tilt 90º, the structural engineer requires a stroke length of greater than 102 inches. A cylinder with a 5.5-inch rod diameter and a 148-inch length is selected. A 5.5-inch diameter piston rod weighs approximately 80.78 pounds per foot. To account for this weight, the engineer must determine if the cylinder requires a non-sag rod. The length of the rod between supports when fully extended will deflect by approximately 0.028 inches with a standard rod [6]. For this application, the deflection value is deemed acceptable. Other design considerations for future iterations include the estimated velocity of travel, break loose pressure values, port sizing for fluid entry and exit, oil consumption estimates per inch of stroke, piping connection methods, seal specifications, the hydraulic fluid, and pump and reservoir sizing. A closed loop electro-hydraulic circuit diagram for synchronizing two lift cylinders with a motor controller (MC) is outlined in the figure below.

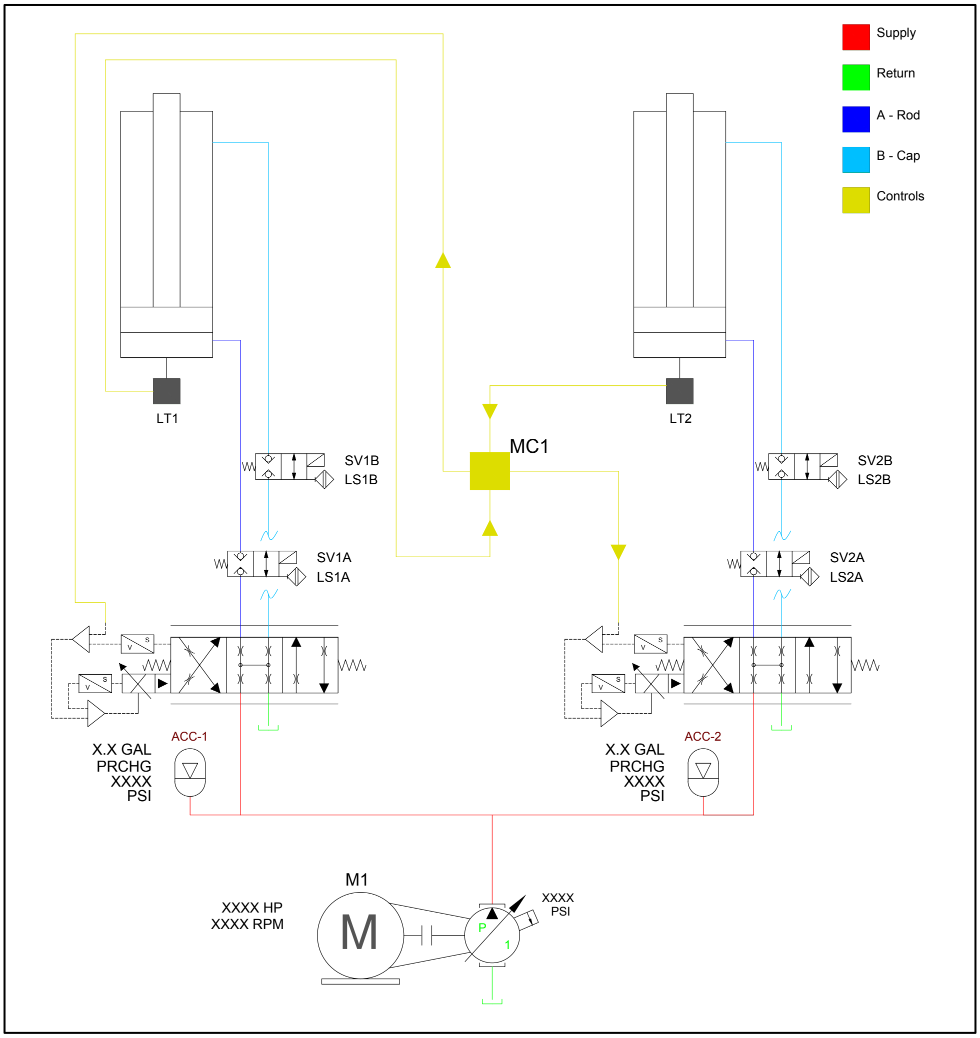


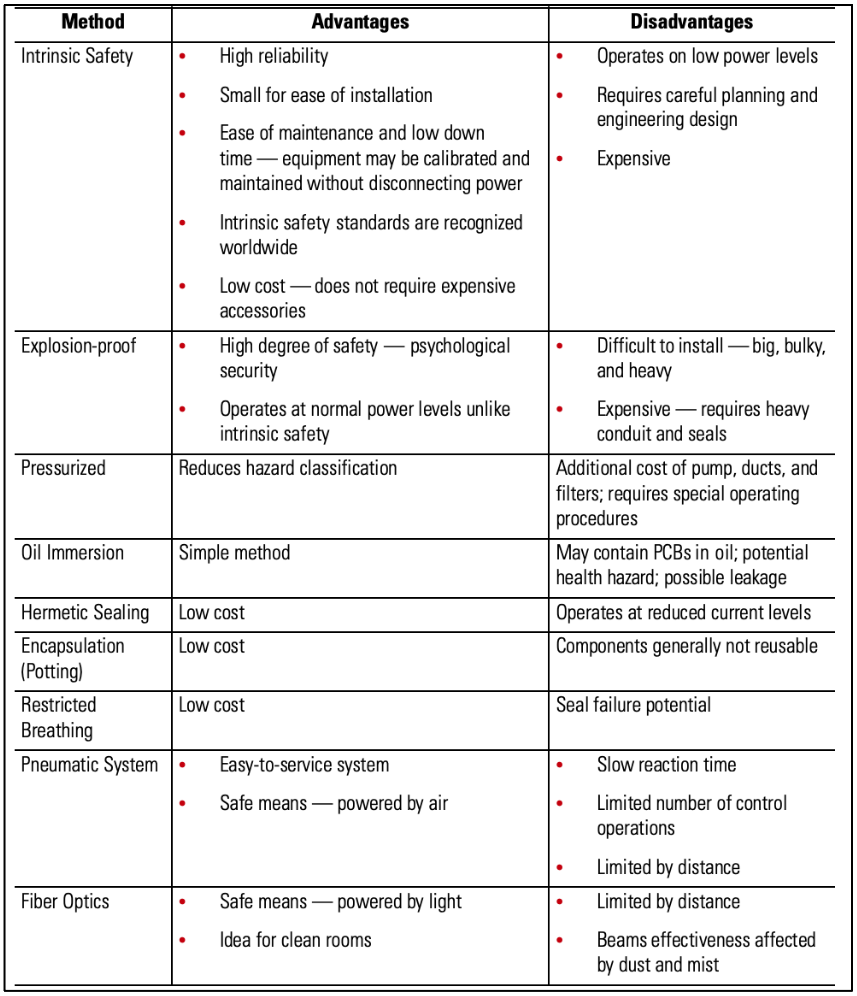
Figure 27: Electro-Hydraulic Closed Loop Control Circuit

3.2.6 Safety System Architecture

The safety subsystem is responsible for ensuring workplace hazards are identified, risks are mitigated, hazard communication programs are established, and standards are well understood and communicated to the team. The safety system is a supporting entity that does not have a specific design by itself but impacts the architectural strategy of other systems. There are numerous hazards present on a launch pad such as electric shock or arc flash, confined spaces, height risks, fires, explosions, and hazardous materials. The occurrence of fires and explosions is a key safety concern at a spaceport and receives the most attention in the form of standards, codes, technical papers, and engineering design. The Occupational Safety and Health Administration (OSHA) is a regulatory body which has established systems classifying locations exhibiting potentially dangerous conditions with respect to the degree of hazard present. OSHA Publication 3073 defines hazardous locations with the following definition: "Hazardous locations are areas where flammable liquids, gases or vapors or combustible dusts exist in sufficient quantities to produce an explosion or fire" [24].

A launch pad is an example of a hazardous location. Due to the dangers, equipment is specially designed and with detailed installation techniques in order to protect against harm to personnel and hardware. Areas are classified as Class I, II or III. Class I is a location where flammable gases or vapors may be present. Class II is defined as a location where combustible dust may potentially be found. Class III is a location where the presence of easily ignitable fiber exists. The three classes are further subdivided into either Division 1 or Division 2. The division represents the likelihood of a flammable concentration present of a specific hazardous material. Materials are placed in a grouping established by the ignition temperature and pressure required for an explosion. Each commodity is classified by Class, Division, and Group [10]. This project assumes the fuel and oxidizer are classified as Class I Division 2 Group D. There is an associated hazard area within these classified areas, driving additional design considerations to ensure hardware is built to operate within this classification. Prior to building a launch pad, a three-dimensional hazard map is generated guiding engineering design decisions such as the location of storage tanks, cross country piping, and hardware selection. Within locations determined Class I Division 2 Group D, the properly rated hardware must be selected to reduce risk of explosion. Table 10 summarizes the protection methods complying with NEC code.

Table 10: Summary of Protection Methods for Hazardous Locations [10]



Method	Advantages	Disadvantages
Intrinsic Safety	• High reliability • Small for ease of installation • Ease of maintenance and low down time — equipment may be calibrated and maintained without disconnecting power • Intrinsic safety standards are recognized worldwide • Low cost — does not require expensive accessories	• Operates on low power levels • Requires careful planning and engineering design • Expensive
Explosion-proof	• High degree of safety — psychological security • Operates at normal power levels unlike intrinsic safety	• Difficult to install — big, bulky, and heavy • Expensive — requires heavy conduit and seals
Pressurized	Reduces hazard classification	Additional cost of pump, ducts, and filters; requires special operating procedures
Oil Immersion	Simple method	May contain PCBs in oil; potential health hazard; possible leakage
Hermetic Sealing	Low cost	Operates at reduced current levels
Encapsulation (Potting)	Low cost	Components generally not reusable
Restricted Breathing	Low cost	Seal failure potential
Pneumatic System	• Easy-to-service system • Safe means — powered by air	• Slow reaction time • Limited number of control operations • Limited by distance
Fiber Optics	• Safe means — powered by light • Idea for clean rooms	• Limited by distance • Beams effectiveness affected by dust and mist

The hazardous area classification map is an artifact generated by the safety group driving launch pad architecture. Along with the hazard map, each risk should be considered and diminished through proper design and workplace practices. Personal protective equipment (PPE), safety clears, fall protection, material safety data sheets (MSDS), employee training programs, Lock-Out Tag-Out (LOTO), and engineering design are examples of risk mitigation along with those outlined in Table 10. In Figure 23, the IBD of system interconnections depicted safety interfaces including a combustible gas detection, emergency stops, structural, fire, and site warning ports. When designing the safety systems on the pad, it is best practice for the SME to have a proper understanding of the risks associated with all systems. For the preliminary design iteration, the safety subsystem will be considered a support system. The design of other disciplines must identify risks and provide mitigation techniques when applicable.

CHAPTER 4

4. REALIZING THE ARCHITECTURAL DESIGN

4.1 Implementation Process

After requirements are developed and an architectural design is generated, the subsequent step is implementation which designs and fabricates each system to conform to the architecture. Inputs are design requirements, verification criteria, and validation criteria, governed by industry standards and safety practices. The outputs are integration constraints, a refined implementation strategy, detailed drawings, updated design documentation, and O&M manuals [49]. In this study, the implementation is fulfilled by generating piping and instrumentation (P&ID) diagrams. These artifacts are predominantly tailored towards the FGSE subsystem and generated by following the MBSE technical processes in Chapter 3.

The purpose of the liquid oxygen loading system is to move LO_2 from the storage tank (ST) to the external tank (ET) on the launch vehicle. LO_2 is stored on the ground in a cylindrical, insulated, double-walled storage tank with a volume of 60,000 gallons (L-3.44). At the ST, a vaporizer maintains ullage pressure by producing gaseous oxygen (GO_2). The ullage is maintained at a higher pressure than in the ET to aid in liquid transfer from the ST to ET. The stages of loading include pad setup, pressurization, cross country chill, slow fill, fast fill, fast fill at reduced pressure and flow, top-off, replenish, and de-tanking [37]. Loading begins by opening the manual blocking valves, setting regulators, and establishing safety clears. The blocking valve and proportional valve upstream of the vaporizer are opened allowing LO_2 to flow through the vaporizer, boiling off and creating GO_2 that is fed back into the ST. While the ST pressurizes, the ET is pressurized using helium fed through a pre-pressurization valve onboard the LV. The first revision of the liquid oxygen subsystem is outlined in Figure 28.

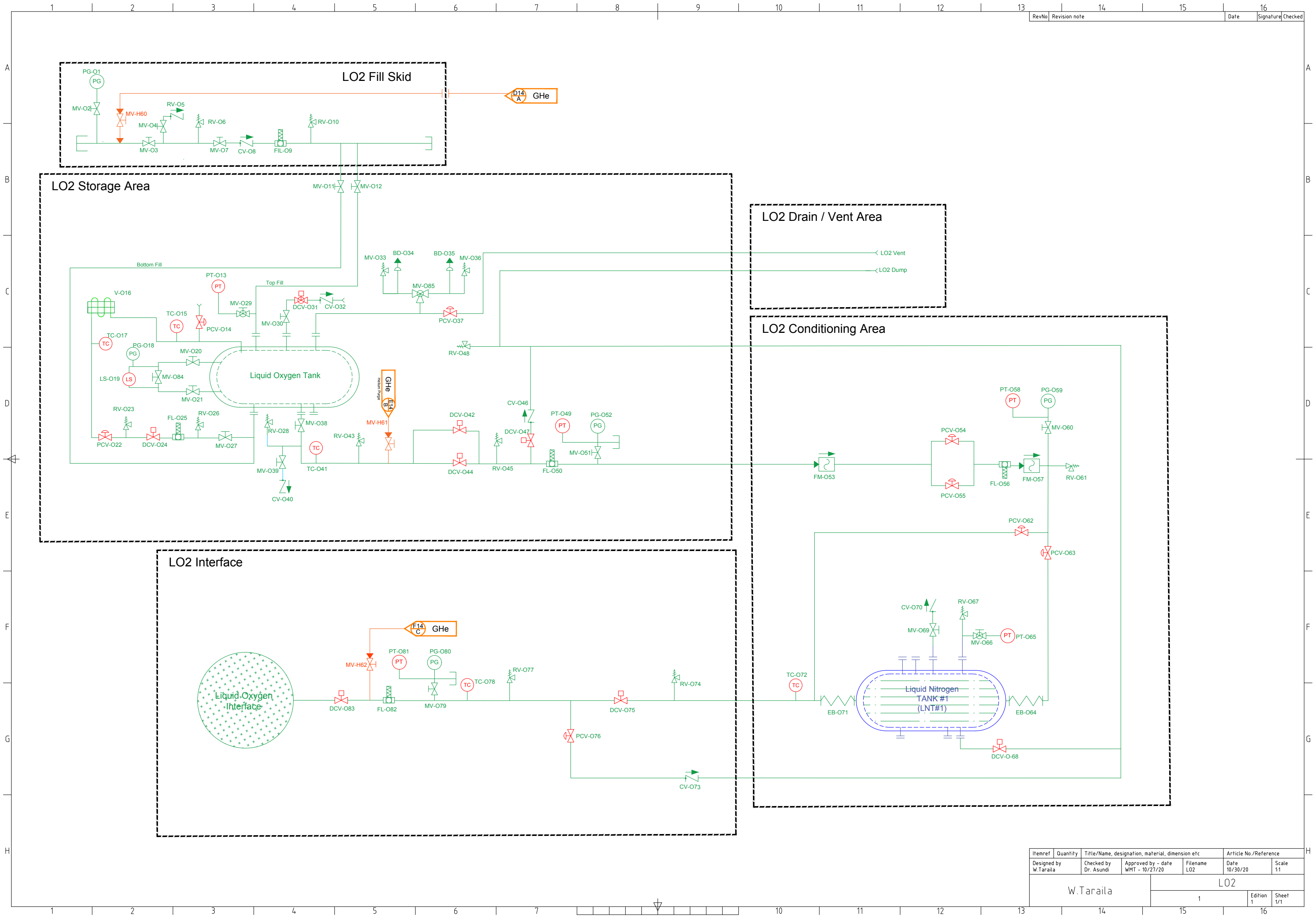


Figure 28: P&ID for FGSE Oxidizer Subsystem

After pressure targets are achieved, the flow through the vaporizer valve and position of the ST vent valve are throttled based on the error measurement between the ullage pressure transducer reading and the ST pressure set point. The cross-country transfer line chills down in stages by introducing LO_2 intro the piping, driven by the ullage pressure of the ST. Operators monitor line temperatures while venting to the drainage area. Once line temperature reaches T_{boil}, (-297.3 ºF) [26], the flow path to the LV is opened. The LV is introduced to LO_2 with the slow fill algorithm, shifting to fast fill when the ET reaches 10% full. The cross-country line has a Coriolis flow meter and two proportional flow control valves in parallel, which use feedback from the flow meter to modulate their commanded position and reach each flow target regime. Loading shifts once more from nominal fast fill to a reduced pressure and flow version of fast fill once the onboard ET reaches 75% full. The ST ullage is reduced, followed by a time delay; then the ground-side flow control valves hunt for the reduced flow rate with the reduced flow state change. When the LV ET reaches 98% of the targeted fill, the top-off cycle begins. The top-off state maintains the LV LO_2 storage tank level between 98% to 100%. The onboard tank is continuously replenished before launch to replace the boil off. If the launch is scrubbed, the LO_2 system de-tanks the LV ET by flowing through an inline filter back to the ground-side ST controlled by the flow control valves and ET ullage. A depiction of an LO_2 loading sequence is shown in Figure 29 [37].

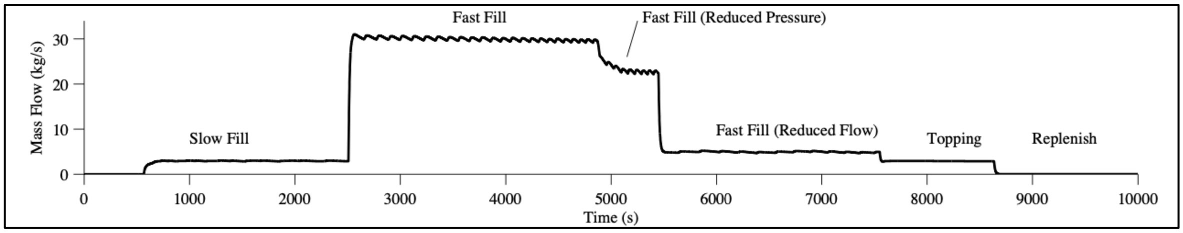


Figure 29: Cryogenic Loading Sequence [37]

The fuel subsystem consists of a fill skid, storage area, flow control area, and launch mount interface. The flow path begins at the fill skid which provides the fuel system the ability to replenish commodity and refill the storage vessel between each mission. The exit port on each relief valve device is plumbed back into the storage tank to prevent inadvertent spilling of hazardous material during fill, circulation, and loading operations. There is a blanket GN2 purge connected to the top of the storage tank, providing ullage head pressure which pushes the fuel towards a pump. A trade study should be performed to determine the outlet velocity of the fuel at the storage tank and the proper size for a fuel pump to meet the interface requirement of 0 to 100 gallons per minute (L-3.40). A circulation loop is added to the P&ID to allow circulation through inline filters prior to loading. This is required in order to accomplish Level-2 requirement (L-2.17) for clean and filtered interface conditions. Pressure, temperature, and moisture sensors at the interface verify fuel requirements are satisfied (L-3.40). Similar to the LO_2 loading sequence, the fuel system has a pressurization, slow fill, fast fill, fast fill (reduced pressure), fast fill (reduced flow), and de-tanking phase. The fuel does not have a top-off cycle because there is no boil off associated with the commodity. The P&ID also shows clear pathways for loading and de-tanking, which are mission Level-1 requirements (L-1.4, L-1.5). The P&ID identifies each component in the fuel system with a unique identifier following a site-wide naming scheme (Part type – commodity #). For example, PCV-F40 is a proportional control valve on the flow control skid within the fueling system with component number forty. Each major part within the bill of materials for the fuel system is represented on the P&ID. In the MBSE SysML model, every component with a unique identifier is a block with unique attributes. In order to fully realize the architectural design, exact part numbers must be found which are suitable, code compliant, and rated for maximum velocity, pressure, and temperature conditions.

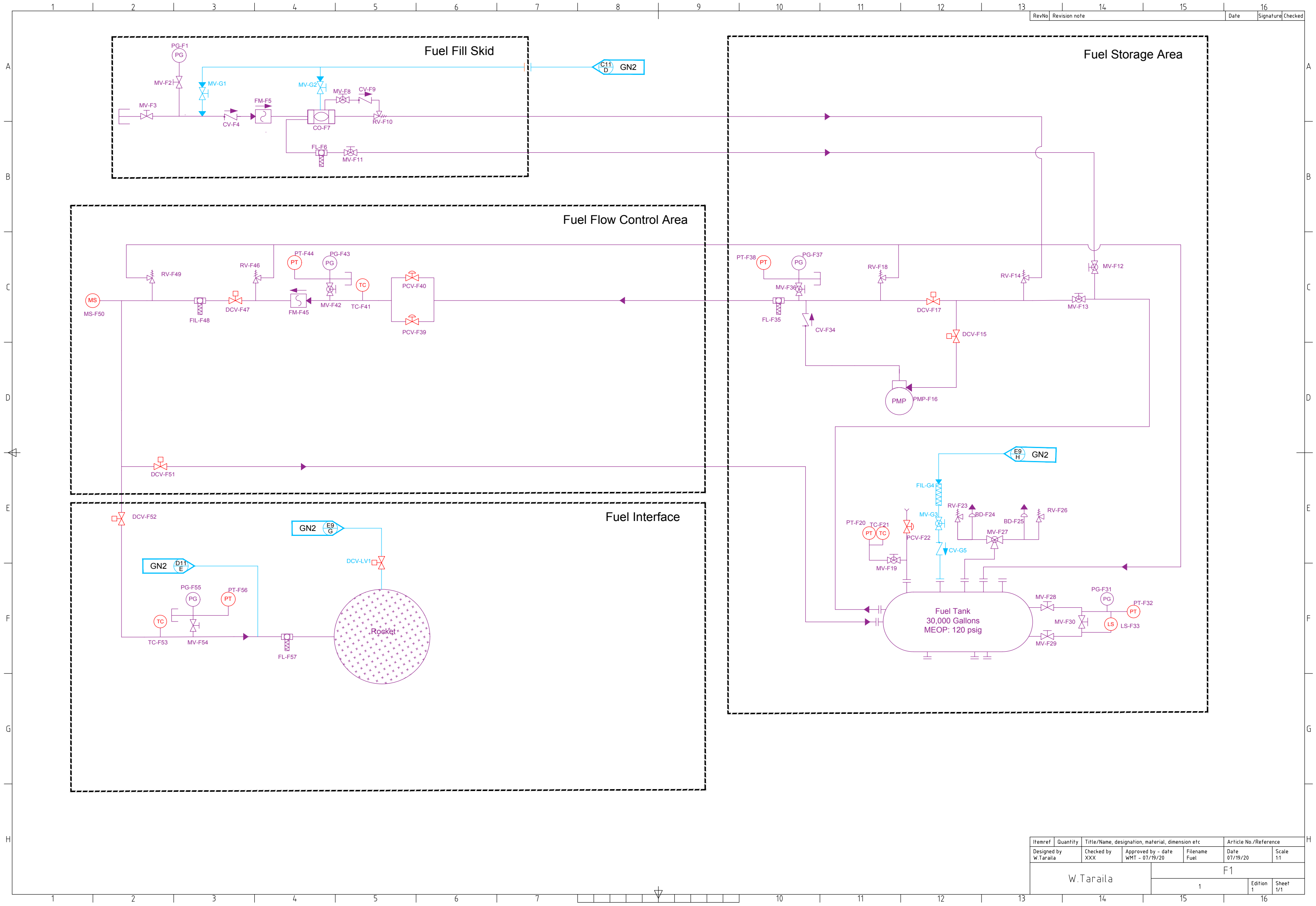


Figure 30: P&ID for FGSE Fuel Subsystem

The nitrogen system consists of a fill skid for replenishment, a storage area, a regulator panel, and seven output legs used for valve actuation, instrumentation purging, pressurization of storage tanks, maintaining cleanliness, and vehicle supply. The nitrogen system is typically active throughout the lifespan of the launch pad in order to support all other system operations and maintain general pad health. In hazardous areas, one of the mitigation techniques deployed is purging and pressurization. This strategy of hazard mitigation requires all electrical control cabinets to be maintained at a positive pressure to prevent potential fire or explosion of devices with electrical energy [10]. The storage area is maintained at 3000 psig, and relief devices are set to discharge at 1.25 times the MEOP. Manual valves are installed on each of the seven service legs, allowing the spaceport to isolate independent legs when their functions are not required. The endpoint of each leg references a separate drawing or vehicle interface. For example, the ‘Fuel Fill Skid Purge’ ties into the Fuel drawing in location ‘A8’ and has a unique identifier of ‘D’. These system interfaces match the proposed system interconnections in the IBD architecture proposed in Chapter 3. The nitrogen subsystem has a Level-3 requirement (L-3.45) to have a storage supply of 100,000 standard cubic feet, provide flowrates to the launch vehicle between zero to one-hundred scfm, withstand a maximum expected operating pressure of five-thousand pounds per square inch, and offer inline filtration of ten micrometers. Each component in the functional P&ID drawing for nitrogen serves a purpose and is linked to a requirement. Prior to construction, the proposed architecture shall be modeled in fluid dynamic simulation software to verify the expected conditions are within the allowable ranges. Hand calculations may also be performed to verify the simulation is properly configured. In future design iterations, a liquid nitrogen system will be tied in to provide the ability to repressurize storage locally.

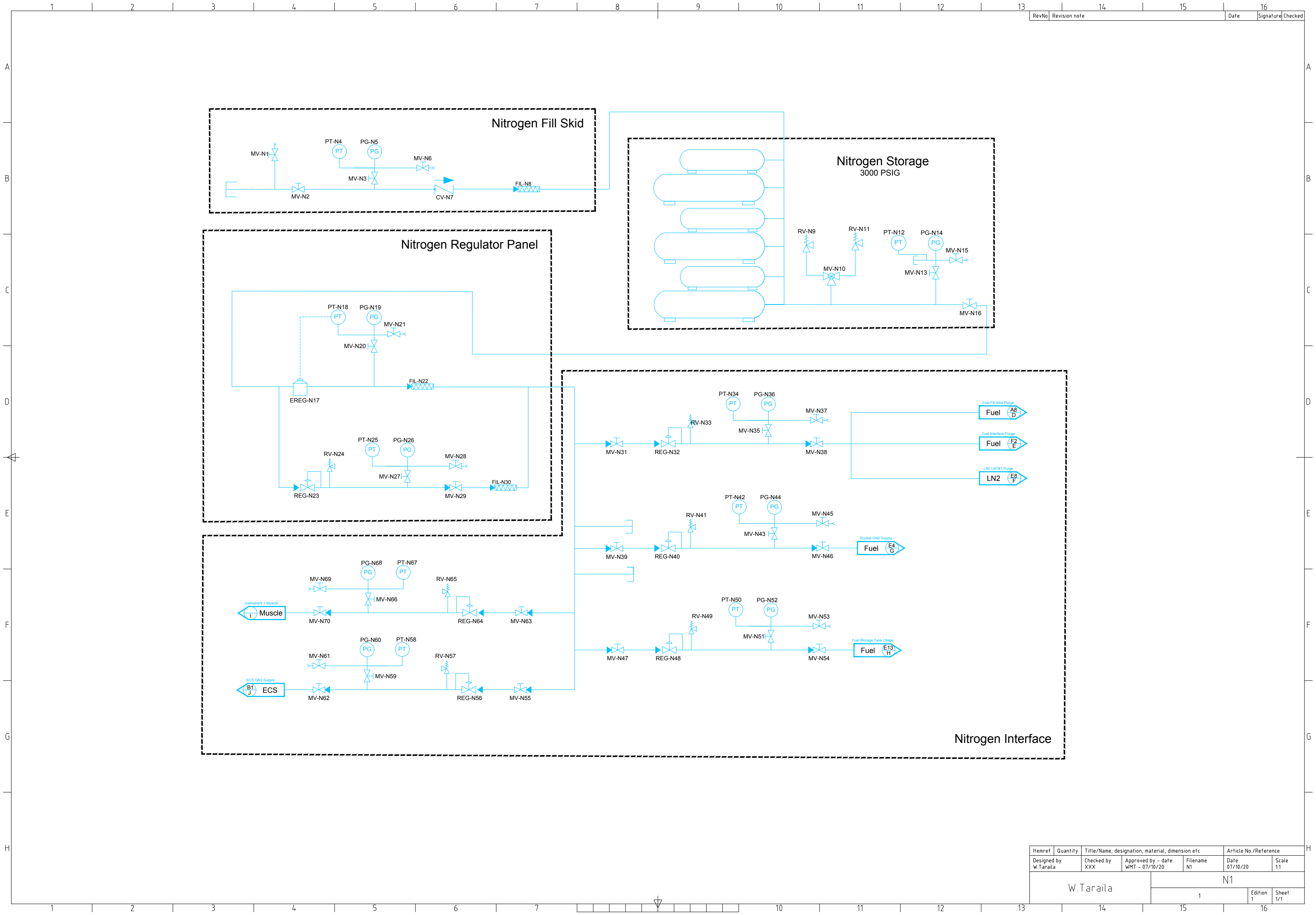


Figure 31: P&ID for FGSE Nitrogen Subsystem

Similar to nitrogen, the gaseous helium subsystem is composed of a fill skid, storage area, regulator panel, and three service legs. The requirements for this subsystem are to provide 100,000 standard cubic feet of storage, deliver flow rates between 0-15 scfm to the LV, be designed for 5,000 psig MEOP, contain 10 μm filters, and purge the LO_2 system. A trade study on the design of the helium system shall be performed using AFT Arrow fluid dynamic simulation software supported by hand calculations. Orifices are installed on the three service lines with different diameters to offer a range of flowrates to the LV and other pad subsystems such as the LO_2 storage tank ullage. A key purpose of helium onboard the rocket is to prevent geysers, which form when heat enters the LO_2 driving the liquid to boil off. The rapid expulsion of boiling LO_2 has the potential to quickly displace large volumes of heavy liquid. As the liquid crashes downward, a water hammer effect is observed. The helium helps to circulate the LO_2, reducing onboard stratification.

Similar to the other commodities discussed, each component on the helium P&ID diagram is tied to a block within the MBSE model and has a unique identifier. An electronic regulator is installed within the helium regulator panel, using an external P&ID feedback loop to more accurately regulate the downstream pressure. The electronic regulator is an example of a helium component that interfaces with the electrical power system and the controls system. EREG-H18 is the unique identifier, and the device monitors the reading of PT-H20 downstream through the controller which reads an analog input signal and derives a command for the electronic regulator based on the delta between actual pressure and the setpoint. Helium is a very expensive commodity with a limited supply. When given the choice of purging equipment, it is typically less expensive to select nitrogen when possible unless the stakeholder has a requirement to use helium.

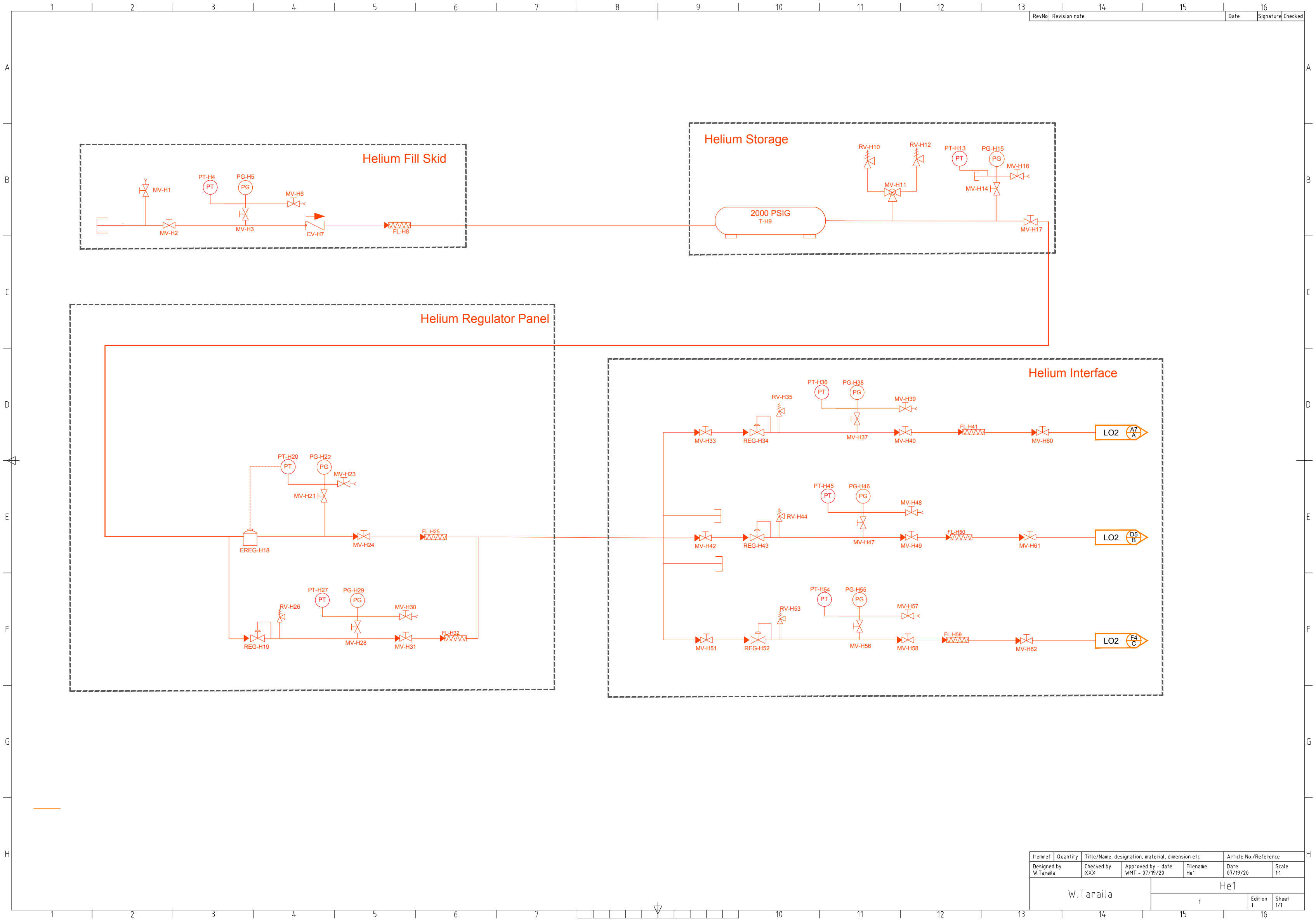


Figure 32: P&ID for FGSE Gaseous Helium Subsystem

For the preliminary launch pad design the liquid nitrogen subsystem is excluded, but provisions are made for future expansion. The proposed design includes a fill skid, storage area, conditioning tank (LNT #1), vaporizer, and pump skid. The storage tank is envisioned to offer many benefits on the pad. The primary use case is for the liquid nitrogen to be converted into gaseous nitrogen, replenishing the launch pad's storage of gaseous nitrogen. The liquid-to-gas expansion ratio of nitrogen is approximately 1:696 at room temperature [13]. As the LN_2 vaporizes, a great amount of force is generated with an enclosed space and further amplified with the use of a pump. Second, the prospective LN_2 conditioning tank shall be designed to have an LO_2 interface, with coils running horizontally through the LN_2 bulk storage. T_{boil} of LN_2 is -320 ºF, offering temperature reduction of the LO_2 supply prior to loading onto the LV. LN_2 may also be utilized to chill down other pad commodities as required.

Other key design exercises include properly sizing the pump which helps to convert LN_2 to GN_2, sizing the vaporizers downstream of the pump, and verifying interface connection points between the multiple subsystems interfacing with LN_2. Each component within the LN_2 P&ID has a unique identifier and is symbolized as a block within the MBSE project. Currently, there are no mission requirements tied to a liquid nitrogen commodity subsystem. Since LN_2 is colder than all other commodities handled on the launch pad, special consideration must be given to selection of hardware within this future system. Cryogenic liquids are a safety hazard with multiple concerns including asphyxiation and oxygen enrichment. While transferring LN_2, the oxygen surrounding the cryogenic containment vessel has the potential to dissolve and create an environment that is oxygen rich. Since the boiling point of oxygen is higher than nitrogen's, LO_2 will evaporate slower than LN_2 and has the potential to collect in a large enough quantity to increase the level of flammability of materials within the surrounding area [26].

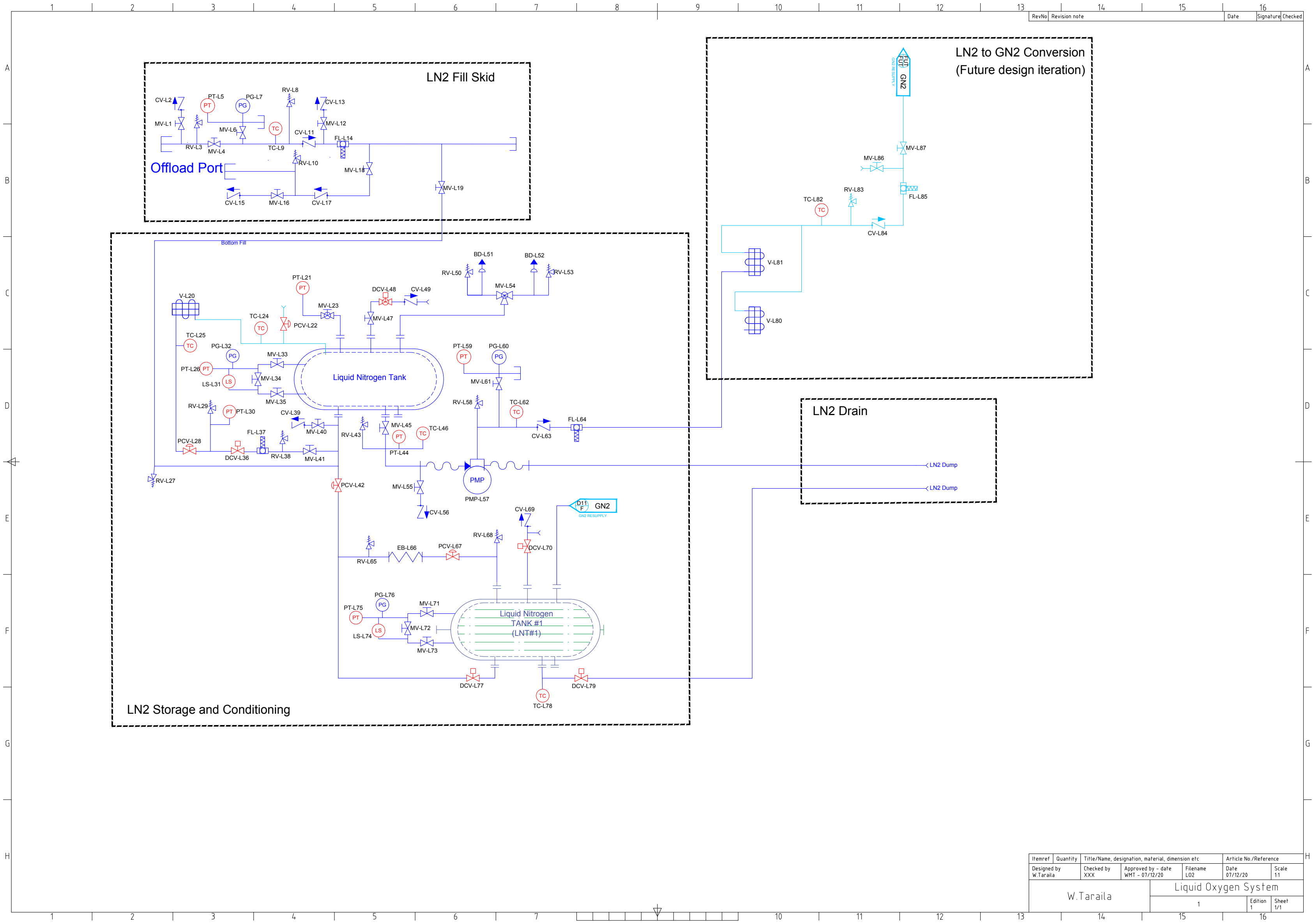


Figure 33: P&ID for FGSE Liquid Nitrogen Subsystem

The HVAC subsystem interfaces with the gaseous nitrogen system. Launch service providers typically require conditioning the key stages of the rocket prior to launch. The stages that require conditioning often include the first stage, second stage, and payload. Parameters such as temperature, filtration, humidity, and pressurization shall be considered and tailored for the specific mission. Depending on ambient conditions, the HVAC subsystem will need to either heat or chill onboard cavities. Challenges of this subsystem include creating an aerospace-suitable control system with parameter ranges wide enough to reduce the adverse effects of the atmosphere. For the preliminary design iteration, the air conditioning system is not included in the analysis. Humidification, cooling, heating, HEPA filtration, protective coatings, and pressurization should be considered when building an environmental control system. Another challenge is to offer environmental conditioning regardless of the orientation of the launch vehicle (horizontal, vertical, transitioning). Prior to fueling, the dew point is typically desired to drop significantly. The system shall be designed to be explosion-proof, operating in a hazardous area. There are also future provisions to interface with two parallel legs, one being an LN_2 conditioning tank and the second being a heater.

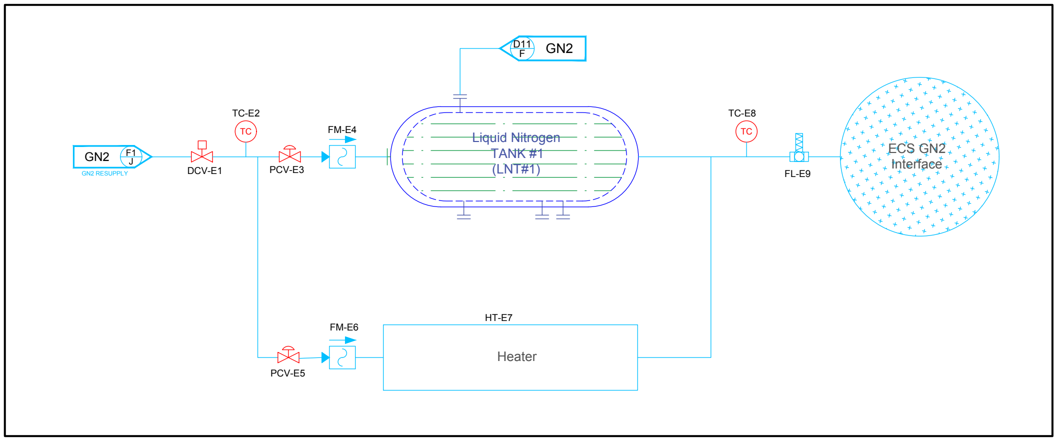


Figure 34: P&ID for FGSE HVAC Subsystem

4.2 Integration Process

Once the implementation phase is complete, the next step is to realize the launch pad by gradually combining elements of the system with respect to the architectural design requirements in the integration process. This is an iterative process which is repeated successively in parallel with the verification and validation processes. According to the INCOSE systems engineering handbook, the inputs include the architectural design requirements, the supplied system elements, and the integration plan. Outputs are a verifiable system, the results of the integration testing, and records of problem resolution as needed. Integration is controlled by predefined agreements, project procedures, and processes. Integration is enabled by enterprise infrastructure, enterprise policies, processes, and standards, and integration enabling systems. Activities performed during integration include defining the integration strategy, receiving system elements, enabling systems per scheduled deliveries, integrating system elements, and recording integration information [49].

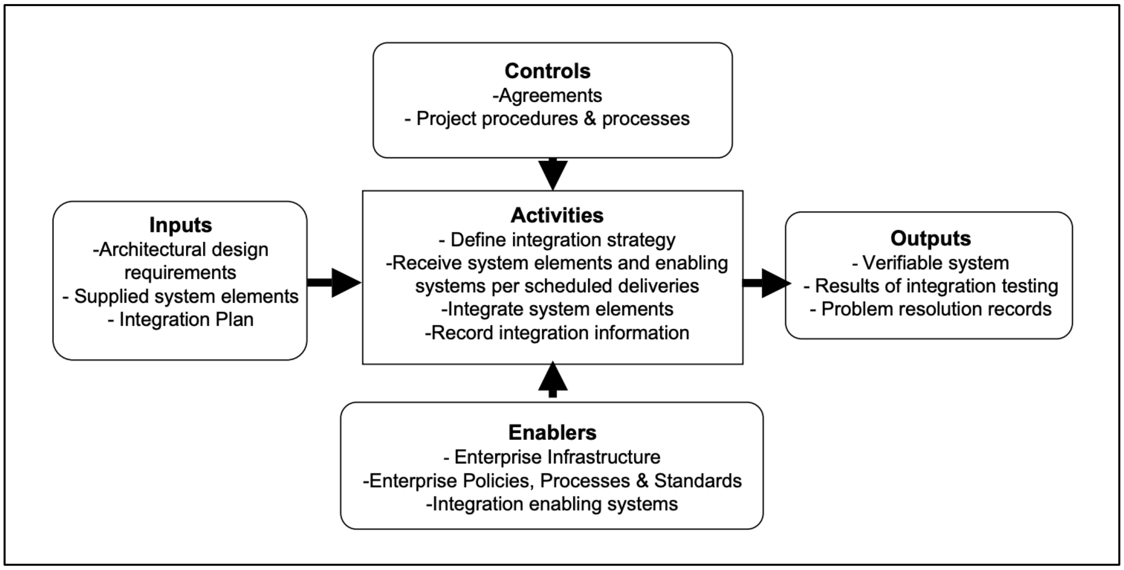


Figure 35: Context Diagram for the Integration Process [49]

4.3 Verification Process

The purpose of the verification process is to verify all mission, system, and subsystem requirements are fulfilled by the proposed system elements and designed system-of-interest. The goal is to confirm that the system has been built right. During this process, a procedure is established for taking remedial actions if non-conformances are found. Inputs are the baseline system requirements, verification criteria, requirements verification traceability matrix, and system elements to be verified. Activities include defining a strategy for system verification, creating and maintaining a requirements verification traceability matrix, and conducting verifications to demonstrate compliance with requirements. Outputs of the verification process are a refined requirements verification traceability matrix and a resulting report of verification and corrective actions taken [49]. There are numerous verification activities to perform for a launch pad. The following section conducts a verification to demonstrate compliance with the helium and fuel subsystem designs proposed in the preliminary design iteration.

4.3.1 Gaseous Helium Design Verification

The design of a pneumatic system presents unique challenges, especially for schedule, which require expedited design approaches. Some of the challenges include providing high quality design work, negotiating during the design review, modeling and analysis that is forgiving so that variable inputs such as evolving requirements are able to be quickly addressed, and designing to a level suitable for procurement early in the design phase to meet schedule. The goal of this exercise is to verify the helium specification (L-3.43) by performing an analysis. Panel line sizing demonstrates that the selected dimensions meet piping circuit pressure drop and flow specifications at the specified conditions (temperature and pressure) while practicing within safe operating conditions with respect to fluid velocity.

Prior to sizing, the SME should have a thorough understanding of the operating modes and functional requirements of the system. This documentation is contained within a panel design specification, datasheets, or requirements document. Some projects evolve at a rapid pace, causing the designer to begin without complete documentation. In this situation, general guidelines should be followed. One example is to follow accepted velocity limits for process fluids. For breathing air or gaseous oxygen, the recommended target maximum fluid velocity is 100 ft/s. Inert gases such as helium or nitrogen should not exceed Mach number 0.2. The pneumatic calculations are driven by performance requirements as well as bounding conditions for flow, temperature, and pressure. Extreme combinations of conditions should be considered, for example minimum inlet pressure and maximum temperature (minimum density) at flow, which would develop the maximum pressure loss through the panel.

The line sizing calculations will be completed using a COTS product called AFT Arrow Software, a recognized flow analysis software package for compressible flow [1]. The software toolkit offers an internal database that is a helpful resource when selecting the tubing and piping characteristics as well as the physical properties of the materials. AFT Arrow contains a wide range of calculation methods, but in this example the "length marching" approach will be implemented wherein the panel piping is divided into a large number of segments which are progressively analyzed while flow conditions and physical properties are adjusted at each step. These calculations are repeated with a Newton-Raphson convergence criterion until the change in calculated conditions between iterations is less than the allowable tolerance specified by the modeler. The AFT Arrow solution is compared versus hand calculations which are accurate at low velocities, and thus low Mach numbers, providing an independent second verification prior to building the system. This results in a higher confidence in the line sizing calculation results.

The hand calculation methodology starts with a known mass flow rate or volumetric flow rate. The mass flow rate and volumetric flow rate may be easily converted with respect to density. This application uses a volumetric flow rate in order to calculate pressure drop and flow velocity. Hand calculations are based on the following equations [12, 14]:

$$Q = \frac{M}{\rho}$$

(1)

$$V = \frac{Q}{A}$$

(2)

$$Re = \rho \cdot V \cdot \frac{L}{\upsilon}$$

(3) Reynolds Number

$$\Delta P = f \cdot \left(\frac{L}{D_h}\right) \cdot \left(\frac{\rho \cdot V^2}{2}\right)$$

(4) Darcy-Weisbach Formula

$$C = \sqrt{\frac{\gamma \cdot R \cdot T}{\mu}}$$

(5) Speed of Sound

where:

Q = volumetric flowrate (ft^3/m)

M = mass flowrate (lb/s)

ρ = density (lbm / ft^3)

V = velocity (ft/s)

A= pipe inner area (ft^2)

Re = Reynolds Number

L = characteristic length (ft)

υ = dynamic viscosity

ΔP = pressure loss (psi)

Mach = Mach Number

D_h = hydraulic diameter

C = speed of sound (ft/s)

γ = specific heat ratio

R = universal gas constant

T = absolute temperature

μ = molar mass of fluid

The requirement for helium is to provide a volumetric flowrate of up to 15 scfm or 0.25 scfs. At 70 °F and 1 atm, helium gas density is 0.0103 lb/ft^3. This equates to 0.002575 lb$_m$/s for M, the mass flowrate. The velocity of the proposed helium system is calculated to verify the limitation requirements are not breached, given a volumetric flowrate and inner pipe cross sectional area. With a volumetric flow rate (Q) of 0.25 ft^3/s and pipe inner cross-sectional area (A) of approximately 0.002478 ft^2, the flow velocity comes out to 100.90 ft/s. Mach number of the fluid at the given conditions is then calculated. One of the limitations that must be considered to assume compressibility factors are negligible is for the velocity of liquid flow to be below 0.10 of Mach and gas flow below 0.20 of Mach. The speed of sound in helium at 70 °F is calculated to be approximately 3311.81 ft/s. The velocity of the helium system must stay below 662.36 ft/s in order to assume compressibility factors are negligible. The calculated Mach number is 0.0305, translating to an acceptable range for assuming incompressible flow. The Reynolds number, a unitless ratio characterizing the flow type under specific conditions, is then found. A Reynolds number lower than 2000 is considered laminar flow, and above 4000 begins the transition to turbulent flow. For this system, laminar flow is desired. The Reynolds number is found to be 4,432.24.

The hand calculation utilizes the Darcy-Weisbach equation for steady state, fully developed, and incompressible flow and compressible flow with limitations. The laminar formula for the Darcy-Weisbach equation is identical to the Hagen-Poiseuille equation, which is analytically derived from the Navier-Stokes equations. The pressure-drop through the pipes due to friction is calculated using the Darcy equation. The Darcy equation can be applied to compressible flows under certain limitations, which have been met by the preceding hand calculations supporting the assumption of incompressible and laminar flow. The final results of the hand calculations are as follows in Table-11. A pressure loss of 2.059 psig is found based on the Darcy-Weisbach formula.

Table 11: Hand Calculation Results - Helium

M	Mass Flow Rate	lb_m/s	Up to 0.002575
L	Pipe length	ft	200.000
γ	Specific heat	N/A	1.667 at 1 atm and 70 ºF
Dh	Hydraulic diameter	ft	0.0562
M(GHe)	Molar mass	u	4.002602
ρ	Density	lb/ft^3	0.0103
Q	Volumetric Flow Rate	ft^3/s	0.25
V	Flow Velocity	ft/s	100.90
C	Speed of Sound (GHe)	ft/s	3,311.81
Mach	Mach number	N/A	0.0305
Re	Reynolds Number	N/A	4,432.24
ΔP	Pressure Loss	$psig$	2.0592

The AFT Arrow flow analysis model was built to represent the P&ID architecture for the GHe System. The flow starts with junction (J1) representing a storage tank at 2,000 psig. The piping is ½ inch diameter throughout the entire model. There are two pressure reducing regulators in parallel (J5 and J34), which join together at a tee directly downstream, but for the initial simulation the J34 regulator is fully closed. Only one of the three of the supply lines are modeled and simulated at a time. There are exit relief valve devices on each line set at 1.25 times the MEOP. The outlet pressures are represented by junctions (J16, J56, and J58) set to atmospheric pressure. There are variable orifice sizes installed on the three independent lines which generate different exit conditions to offer various ranges of flow. Outlet #1 (J16) has a 0.10-inch orifice to deliver a volumetric flow rate of 14.8841 scfm. Outlet #2 (J56) and outlet #3 (J58) are simulated closed to focus on the first helium interface. The process may be repeated for the remaining outlets. The size of the orifice may also be adjusted to fine tune the delivered flow rate at the interface. The AFT flow analysis output is within family of the requirement of 15 scfm and the hand calculations for flow velocity, Mach number, and Reynolds number.

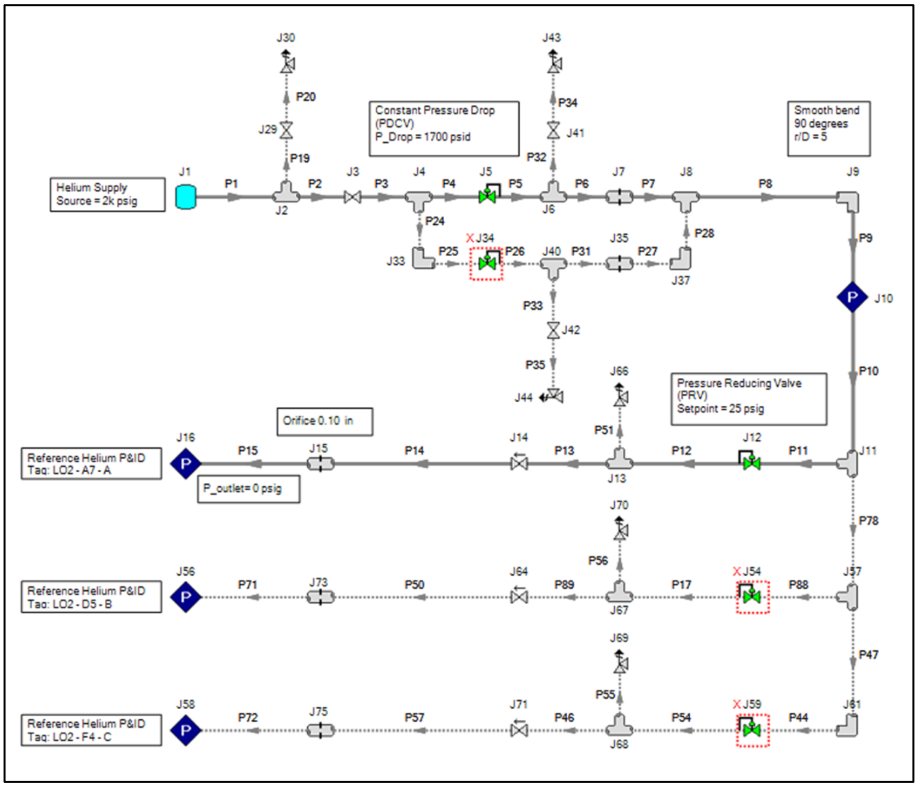
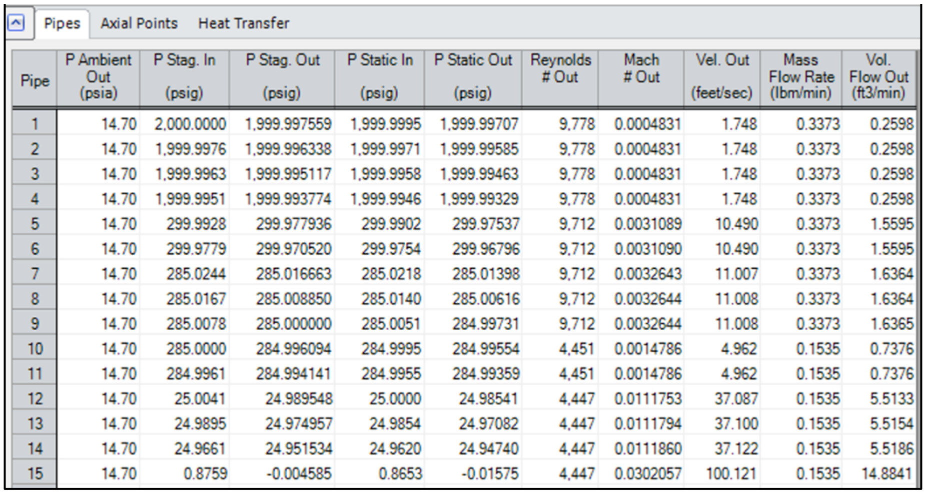


Figure 36: AFT Arrow FGSE Helium Fluid Dynamics Model Setup

The results of the pneumatic flow simulation are outlined in Figure-37. Key parameters for review are exit velocity, Mach number, mass flow rate, volumetric flow rate, Reynolds number, and stagnation pressure. The path from J1 (source) to J16 (outlet #1) is plotted for the main parameters of interest, displaying how key variables fluctuate throughout the system. The path from J1 to J56 (outlet #2) and J1 to J58 (outlet #3) are not graphed, but the analysis process is similar in nature. The verification of requirement (L-3.43) is partially fulfilled with the hand calculations and subsequent flow analysis using COTS software. Confidence that the proposed design will meet stakeholder needs is increased with the exercise and datasets attached.

Table 12: AFT Arrow Helium Trade Study Outputs



Pipe	P Ambient Out (psia)	P Stag. In (psig)	P Stag. Out (psig)	P Static In (psig)	P Static Out (psig)	Reynolds # Out	Mach # Out	Vel. Out (feet/sec)	Mass Flow Rate (lbm/min)	Vol. Flow Out (ft3/min)
1	14.70	2,000.0000	1,999.997559	1,999.9995	1,999.99707	9,778	0.0004831	1.748	0.3373	0.2598
2	14.70	1,999.9976	1,999.996338	1,999.9971	1,999.99585	9,778	0.0004831	1.748	0.3373	0.2598
3	14.70	1,999.9963	1,999.995117	1,999.9958	1,999.99463	9,778	0.0004831	1.748	0.3373	0.2598
4	14.70	1,999.9951	1,999.993774	1,999.9946	1,999.99329	9,778	0.0004831	1.748	0.3373	0.2598
5	14.70	299.9928	299.977936	299.9902	299.97537	9,712	0.0031089	10.490	0.3373	1.5595
6	14.70	299.9779	299.970520	299.9754	299.96796	9,712	0.0031090	10.490	0.3373	1.5595
7	14.70	285.0244	285.016663	285.0218	285.01398	9,712	0.0032643	11.007	0.3373	1.6364
8	14.70	285.0167	285.008850	285.0140	285.00616	9,712	0.0032644	11.008	0.3373	1.6364
9	14.70	285.0078	285.000000	285.0051	284.99731	9,712	0.0032644	11.008	0.3373	1.6365
10	14.70	285.0000	284.996094	284.9995	284.99554	4,451	0.0014786	4.962	0.1535	0.7376
11	14.70	284.9961	284.994141	284.9955	284.99359	4,451	0.0014786	4.962	0.1535	0.7376
12	14.70	25.0041	24.989548	25.0000	24.98541	4,447	0.0111753	37.087	0.1535	5.5133
13	14.70	24.9895	24.974957	24.9854	24.97082	4,447	0.0111794	37.100	0.1535	5.5154
14	14.70	24.9661	24.951534	24.9620	24.94740	4,447	0.0111860	37.122	0.1535	5.5186
15	14.70	0.8759	-0.004585	0.8653	-0.01575	4,447	0.0302057	100.121	0.1535	14.8841

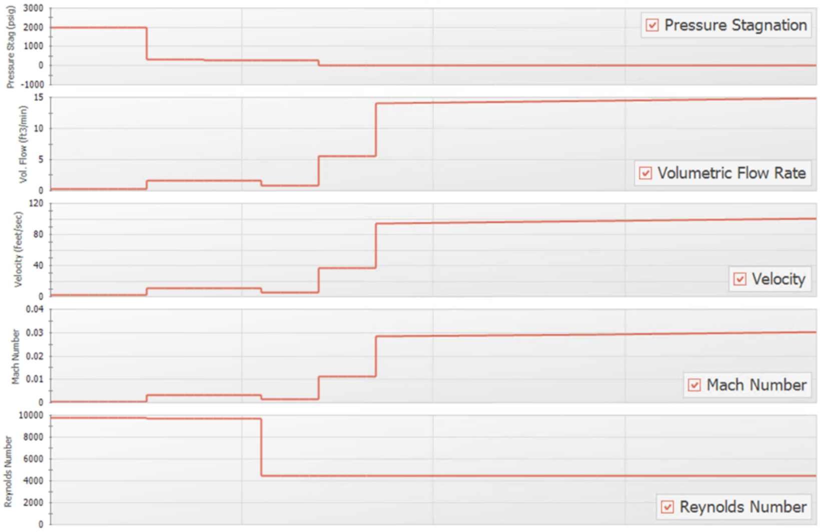
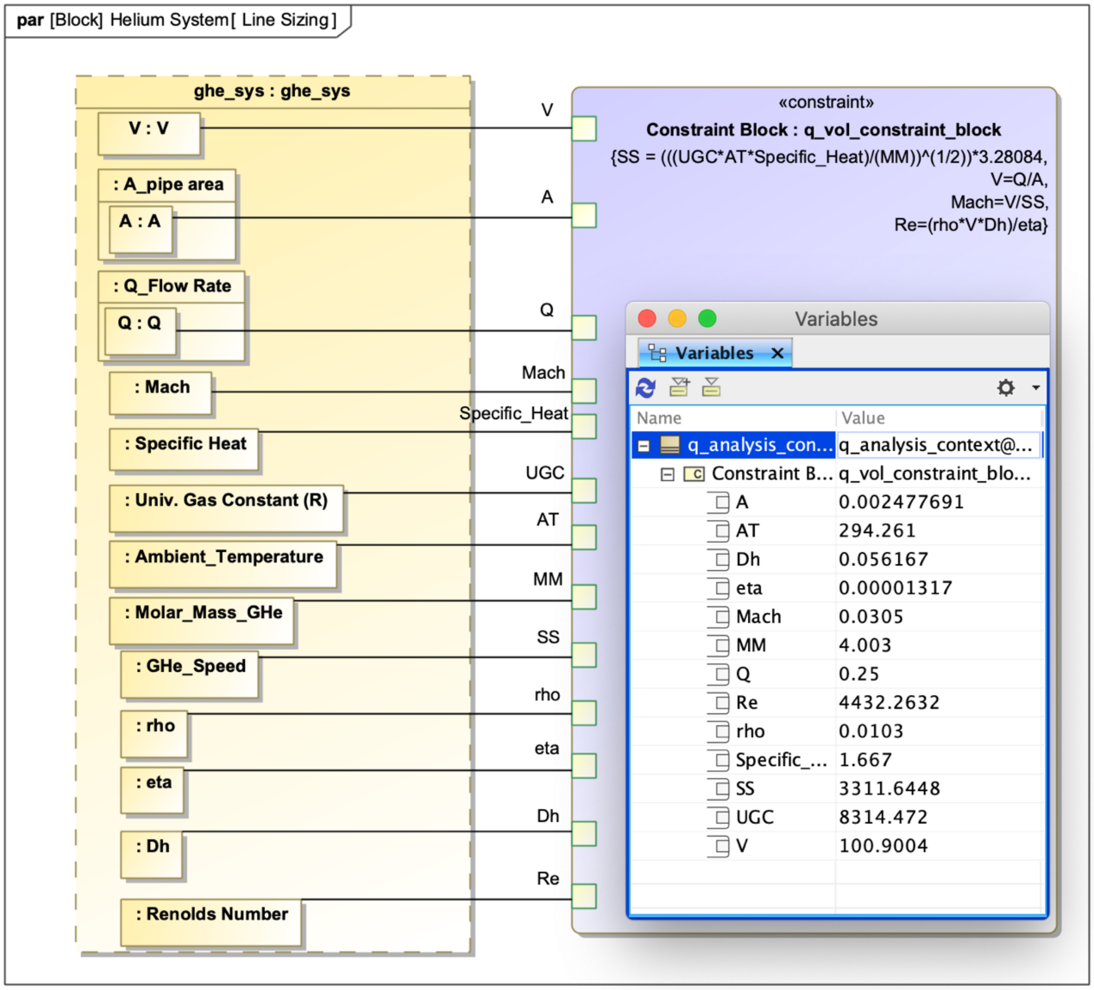


Figure 37: AFT Arrow FGSE GHe Fluid Dynamics Model Output

The helium line sizing exercise outlined the process for a system engineer to design a pneumatic panel to meet Level-3 subsystem requirements. A nitrogen line sizing exercise would follow a similar analysis and design strategy. Due to the similarity of the two trade studies, a nitrogen trade study will not be performed. Next steps are to integrate the hand calculations and outputs of the simulation into the FGSE package within the SysML model via a parametric diagram, shown in the following figure. The requirements should be traced to the specific blocks and parametric diagrams representing the analyses performed.



Name	Value
q_analysis_con...	q_analysis_context@...
Constraint B...	q_vol_constraint_blo...
A	0.002477691
AT	294.261
Dh	0.056167
eta	0.00001317
Mach	0.0305
MM	4.003
Q	0.25
Re	4432.2632
rho	0.0103
Specific_...	1.667
SS	3311.6448
UGC	8314.472
V	100.9004

Figure 38: Parametric Diagram for Helium System Line Sizing

4.3.2 Fuel Design Verification

The fuel system requirement is detailed in the P&ID collection specifying a 30,000-gallon storage tank, MEOP of 150 psig, and an interface flow rate of 0-100 gallons per minute with a nitrogen purge interfacing with the fuel subsystem. To meet this requirement, the fuel subsystem P&ID is simplified and represented by Figure 39 [37].

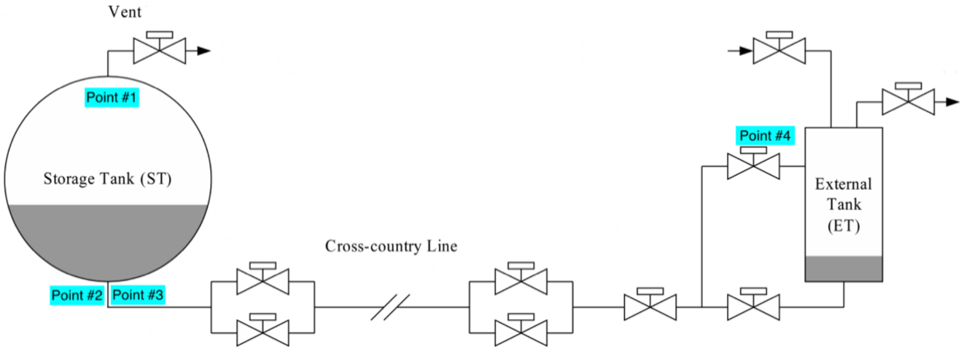


Figure 39: Fuel Analysis Model Setup

Bernoulli's principle is applied using the conservation of energy applied to a flowing fluid while assuming steady state, incompressible, inviscid flow along a streamline. The equation states that in regions where the flow velocity is increased, the fluid pressure in this region decreases [12].

$$P_1 + \frac{1}{2}\rho {v_1}^2 + \rho g h_1 = P_2 + \frac{1}{2}\rho {v_2}^2 + \rho g h_2$$

The summation of the pressure energy, kinetic energy per unit volume, and potential energy per unit volume at two points in a flow path must be equal. Point #1 is located at the top of the storage tank where pressure (P_1) is equal to the ullage pressure on the tank, the velocity is zero, and at the top of the storage tank is height (H1). Point #2 is chosen to be at the outlet at the bottom of the tank. The flow velocity at the outlet of the tank is found from these two points using Bernoulli's equation with a known outlet diameter resulting in a volumetric flow rate (Q) in gallons per minute. The resulting Reynolds number indicates that the flow is in the transition zone from laminar to turbulent. Using the Reynolds number and relative roughness of Schedule 40 pipe with a 1.0'' diameter, the friction factor (f) for 1.0-inch clean commercial steel pipe in the turbulent flow regime is found. The head loss when flowing through valves and fittings is generally represented in terms of the resistance coefficient (K). This static head loss, or equivalent length of pipe diameters (L/D) which will cause the same head loss as the valve, is referred to in terms of velocity head [12].

$$R_e = \frac{Q\rho}{d\mu}$$

$$K_{pipe} = f\frac{L}{D} \qquad K_{90^{\circ}\,elbow} = 14f_T \qquad K_{ball\ valve} = 3f_T$$

$$K_{bend} = (n-1)\left(0.25\pi f_T\frac{r}{d} + 0.5K\right) + K$$

$$K_{Total} = \sum_{i=1}^{n} K_i$$

$$h_L = K\frac{v^2}{2g} \qquad H = h_{interface} + h_L$$

$$P_{outlet} = \rho(h_1 - h_2 + \frac{P_1}{\rho} - h_L)$$

The resistance coefficients for the pipe, elbows, and ball valves are summed together to find K_{Total}, which is an input to calculate the final head loss at the interface. The interface is at a higher elevation than the storage tank and cross-country line, increasing the pressure drop between the storage area and the interface. The fuel system is designed to have the ability to meet flow rates at the interface using storage tank ullage pressure (GN_2) or a pump. The pump is sized based on pump head, flow rate, and brake horsepower (bhp) requirements. The following table summarizes the inputs and calculated outputs of the fueling subsystem analysis model.

$$E_p = \frac{v_2}{g} + f\frac{l}{d}\frac{{v_2}^2}{2g} + h_{pump} + h_L \quad (feet)$$

$$P_{size} = \frac{(SG_{fuel})*Q*E_p}{\eta_{pump}} \qquad bhp = \frac{Q*H*\rho}{\eta_{pump}}$$

Table 13: Fuel Hand Calculations Summary

P1	Pressure Point #1 – Top of tan	psig	7
V1	Velocity Point #1 – Top of tank, zero velocity	Ft/s	0
H1	Height Point #1 – Top of tank	Ft	10
P2	Pressure Point #2 – Outlet of tank, case #1	Psig	0
V2	Velocity Point #2 – Outlet of tank, case #1	Ft/s	41.44
H2	Height Point #2 – Outlet of tank, case #1	Ft	2
P3	Pressure Point #3 – Outlet of tank, case #2	Psig	9.85
V3	Velocity Point #3 – Outlet of tank, case #2	Ft/s	41.44
H3	Height Point #3 – Outlet of tank, case #2	Ft	2
P4	Pressure Point #4 – Interface / outlet, case #2	Psig	2.97
V4	Velocity Point #4 – Interface / outlet, case #2	Ft/s	41.44
H4	Height Point #4 – Interface / outlet, case #2	Ft	10
d	Pipe Diameter	In	1.0
A1	Pipe Area	In^2	0.79
ρ_{water}	Density of water	lb/ft^3	62.24
μ	Dynamic viscosity	cP	1.64
f_T	Friction factor, Schedule 40 - 1.0 in.	N/A	0.023
l	Length of pipe	ft	200
SG	Specific Gravity of Fuel	N/A	0.8226
Q	Volumetric Flow Rate	gpm	103.34
Re	Reynolds Number	N/A	163,231
K_{pipe}	Pipe resistance coefficient	N/A	55.20
$K_{90^{\circ}\,elbow}$	90° elbow resistance coefficient (11 total)	N/A	0.3220
$K_{ball\ valve}$	Ball valve (1.0 in.) resistance coefficient	N/A	0.0690

K_{Total}	Total resistance coefficient	N/A	59.018
h_L	Head loss	ft	1632.24
η_{pump}	Efficiency of the pump	N/A	0.90
h_{pump}	Height of the pump	ft	2
E_p	Pump head required	ft	372.23
P_{size}	Pump size	kW	3.21
bhp	Break horsepower	Hp	18.91

The following outlet pressure (P4) was derived without the use of a pump. In future design iterations, next steps would be to incorporate the pump into the interface pressure calculation and build a parametric diagram of the architecture verification strategy within SysML.

In addition to the analysis method, inspection, demonstration, and testing are also examples of verification methods. Inspection involves the examination of a system by using the five senses (see, hear, touch, taste, smell) and can include a measurement or physical manipulation. An example of a requirement that is checked off by means of inspection could be the installation, connection, and display of a particular camera view broadcasting over the network. If the operator can inspect the view and definitively say that the camera view meets the requirement, then this item may be closed via sight. Demonstration of a requirement entails manipulating the system as intended in order to prove that the result of the demo occurs as expected. For example, in a software application a user may enter all of the required fields or inputs verifying that the report generated returns the datatype required. Testing verifies a system by incorporating a predefined or coordinated series of data, inputs, or stimuli in order to guarantee that the system will yield a specific output which is specified by the requirements. An example of test verification would be flowing LO_2 to the interface of the launch mount and measuring the outlet temperature with an RTD to prove the range is within the expected temperature bands. The fourth method of requirement verification, analysis, was applied to certify a subset of the requirements for the fuel and helium subsystem by means of calculations and models. An analysis allows the operator to construct predictive statements about the characteristic behavior of a system based on a sample's test results that have already been confirmed or by merging the result of individual tests to determine a new quality about the system. Analysis is also used to predict failure of systems by incorporating nondestructive testing to deduce failure points. In the future, the MBSE model will be applied to assist in the definition of the verification methods and the implementation of procedures traceable to requirements.

4.4 Transition and Validation Processes

The process of transition entails the transfer of custody and responsibility from the development team to the operational and support organizations. Inputs to transition are the system-of-interest, installation plans, and a prepared operational environment. During transition, activities include the preparation of installation procedures, site operational preparation, site installation and construction, and the documentation of results or anomalies. Outputs of the transition process are refined installation procedures, results from the final acceptance activities, and an installed system [49].

Following a successful transition, the validation process confirms that the realized system complies with the defined requirements of the stakeholders. Validation of the system is subject to the approval of the key stakeholders and project authority. Validation ensures that the right system has been built. Inputs to validation are an integrated system that has been released for validation and criteria for validation of stakeholder requirements. Activities performed during validation include definition of validation procedures, ensuring system readiness, demonstration of conformance to stakeholder requirements, recommendations for corrective actions, and the attainment of stakeholder acceptance. Outputs of validation are the validation procedures, the reported results of validation activities, and a listing of corrective actions [49].

4.5 Operation and Maintenance Processes

The operational process uses the system to deliver the intended services. The maintenance process is typically performed concurrently with operations. Inputs to operation include an accepted system, operational procedures, and consumables. Activities performed during operation are the execution of the concept of operations, maintaining a qualified staff, obtaining consumable materials, monitoring operations, assessing performance, reporting system malfunctions, and collecting operator and stakeholder feedback. Outputs of the operational phase are the ongoing system services, the results of monitoring system performance, and recommendations for corrective actions. The life cycle phase, called the utilization stage, corresponds to the operations process.

The maintenance process helps to sustain the system and extend the lifetime to support ongoing operations. Inputs of maintenance are an accepted system, maintenance procedures, spare parts and consumables. During maintenance of a launch pad, activities include the refinement of the maintenance strategy, definition of design constraints imposed by maintenance, the implementation of maintenance and logistics support procedures, the performance of maintenance actions, and documentation of work steps performed. Outputs of this process are a refined maintenance strategy, ongoing system services, logistics service records, and maintenance historical records. The maintenance process relates to the support stage within the list of life cycle phases [49].

4.6 Disposal Process

The disposal process is the final step of the ISO/IEC 15288 technical process guide, and the purpose of this step is to remove a system element or the entire system from the operational environment with the intent of concluding its use. Disposal also deals with any hazardous materials and or waste products associated with the system termination in accordance with all applicable regulations, statutes, policies, and applicable guidance. Inputs of disposal are a depleted system element, production and operational environments, and a system disposal plan. Activities during disposal include the refinement of the disposal strategy, imposition of disposal constrains on requirements, system deactivation, system removal, and the maintenance of historical archives documenting the disposal process. Outputs are the final disposal strategy, the disposal constraints on requirements, refined system elements, and documentation as required [49].

CHAPTER 5

5. CONCLUSION

This thesis has described a model-based systems engineering (MBSE) technique for designing a small-lift launch pad while following the ISO/IEC 15288 technical method. In Chapter 1, the main objective stated was to display the benefits of designing a complicated system with multiple disciplines in a model-based atmosphere. The single source of truth living within the MBSE model integrates all of the project's technical information using SysML. The paper identified advantages of the MBSE method. By using the nine types of SysML diagrams, the ability to increase precision of system specifications within the model and reduce the propagation of errors was demonstrated. The paper described how MBSE offers improvements with respect to traceability while managing the requirements, design, analysis, and verification. The system model's capability to be maintained and continue evolving with an everchanging set of system design specifications throughout the different life cycle stages was also discussed. If implemented correctly, the notion that an MBSE model may support multiple projects and potential reuse in future projects was identified. With the tools discussed, the fundamental understanding of all aspects of a system are better captured aiding in the reduction of miscommunication and confusion among the stakeholders and development team.

In Chapter 2, a literature review was performed, covering a historical record of rocket launch sites in the United States, system engineering models, and an overview of SysML. The knowledge gained from reviewing MBSE techniques and SysML was incorporated into examples of all nine types of SysML diagrams tailored to the system of interest. A package structure was proposed to organize the launch pad project while considering the different engineering disciplines required. The requirements, structure, use cases, and behavior packages comprised the top-level mission organization. As the model's architecture matured, technical details for each discipline began to grow with links tracing to other subsystems. The disciplines discussed included civil, controls, FGSE, software, safety, electrical power, mechanical, and hydraulic subsystems. Another objective of this research was to increase awareness regarding "locked in" life cycle costs early in the design and development stages. With the use of the MBSE model, an advantage is the identification and resolution of problems early to reduce cost and engineering rework.

Using SysML, multiple aspects of the system were described. The breakdown of individual systems was represented as a hierarchy of subsystems and components. Elemental interconnections were logically linked using object-oriented software. Examples of behavior, actions, inputs, outputs, and control flows of individual systems, subsystems, and components were given. Sequences of message exchanges, such as a loop for automating the ullage pressure in a cryogenic tank were capable of being represented in SysML diagrams representing information exchange between parts. The current state and transitions to alternative states of blocks, systems, or components were demonstrated. An example of state representation was an electropneumatic control valve which transitioned from the normally open state to the closed state when the controller commanded the end device. Future additions to the launch pad model shall include top level states, such as a launch pad standby state where the entire system verifies the state of all end devices in every subsystem and reports state status back to the controller. Block properties were identified using internal attributes. Examples of element properties included the range of a sensor, the hazardous mitigation technique, or the maximum expected operating pressure of a component installed on a pneumatic pipe. The text-based requirements were discussed in detail with an example set of requirements for a small class launch pad proposed in Chapter 3. The mission, system, subsystem, and component-level engineering design to meet a specific requirement was examined in both a requirement diagram form as well as a requirements table. The traceability relationships offered within an MBSE model were explored and an example of a high-pressure nitrogen requirement diagram was given, identifying methods of increased traceability. These requirements were analyzed with the end goal of realizing an acceptable architectural design.

With the ISO/IEC 15288 technical process as the framework and SysML as the medium, the launch pad's development within MBSE began to build traction in Chapter 3. The first step of the technical process was to plan the modeling effort. The model's objectives, scope, milestones, deliverables, software selection, and modeling methodology were reviewed in detail. Artifacts were linked to schedule milestones, giving the project a path forward with expected due dates for key events and resources. The second step in the technical process was accomplished by analyzing the needs of the mission and the stakeholders. The mission objectives were defined, and table of requirements were proposed. Measures of effectiveness (MoE) were introduced as a means of gauging whether objectives were achieved. The mission requirements tables were a key resource referenced throughout the remainder of the research paper, linking specific requirements to proposed designs. Future work for the launch pad MBSE model includes expanding upon the mission requirements and adding further detail to each subsystem. The requirements set forth were intended to give an example of necessities to launch a rocket, but in reality, the table would be orders of magnitude larger. Based on the requirement tables for Level-1 (mission), Level-2 (system), and Level-3 (subsystem), Internal Block Diagrams (IBD) were created to begin synthesizing an architectural foundation. The IBD was applied at the system level to show interfaces between engineering disciplines, and then further decomposition via IBD diagram detailed the interconnections of a particular system.

The architectural design foundation in turn assisted with the implementation in Chapter 4. Piping and instrumentation (P&ID) diagrams were built for FGSE systems, and a hydraulic circuit diagram was generated based on the requirements tables. These diagrams gave the reader a path to follow from the source to final interface, with a proposed list of inline components. Each component on the P&ID was linked to a requirement and served a specific purpose. With respect to the P&ID artifacts, a future step shall be refining the diagrams in succeeding iterations. In addition to P&ID refinement, the engineering team shall add more P&IDs that were not included such as a water deluge drawing and a more intricate environmental control system (ECS) drawing. The MBSE model is capable of capturing all of the information offered in the P&ID and building upon this with many other features. For example, information about each component such as in the bill of materials (BOM), controls information such as channel type and slot number, requirement number, and parametric analysis information may all be linked to offer the system engineer a particularly valuable resource during the life of the project. The document-based approach would require at least five separate documents to ascertain all of the information tied to a single block in SysML.

A rudimentary mission timeline was proposed and introduced as an activity diagram. Key mission activities such as sampling, purging, regulator setups, and final loading were outlined, starting at thirty days prior to a launch and finishing at takeoff. A preliminary set of top-level mission failure modes were introduced, which would cause a loss of mission. The mission level depictions aid in mitigation of project failure. Black box specifications for a launch pad were introduced as a way to reflect functional, performance, physical, and interfacial mission requirements. Using the correct level of abstraction, alternative concepts are capable of examination with the assistance of this black box concept. Technical performance measures (TPM) were applied to gauge system-level performance while measures of effectiveness (MOE) were applied to meter mission-level performance.

The launch pad was further decomposed into individual parts and analyzed to perform verification by analysis. The design process is iterative in nature and a first analysis of multiple launch pad design choices was attempted. Trade studies on the gaseous helium subsystem helped the designer to determine the appropriate size of piping, length of piping, and inline components which would produce less tempestuous outlet conditions. Hand calculations and COTS fluid dynamics software were used to design a pneumatic system to meet specific interface requirements. General design guidelines were provided, giving future engineers a starting point and process to follow. Conveyance of the trade study within the SysML model is beneficial for the design team. The fuel subsystem’s flow analysis was a second trade study that offered general design techniques to follow and parametric relationships to build into the SysML model. Bernoulli’s principal was applied at different segments of the fueling system in order to determine volumetric flow rate, pressure drop, Reynolds number, and potential pump characteristics needed to achieve interface flow conditions. Designers’ next steps would be to propose specific hardware that accepts the model’s inputs and delivers interface outputs.

The safety subsystem was discussed briefly, highlighting protection methods for hazardous locations. This is a fitting example of how MBSE makes design details easier to track. With hundreds of end devices and multiple protection methods available, a block attribute representing this characteristic for each end device promotes an interdisciplinary understanding of the entire facility. Each method of protection has implications on other subsystems such as additional inline electrical components for intrinsically safe devices, specific installation instructions for explosion-proof hardware, or pneumatic tubing routed to pressurized enclosures. Individual constituents of the electrical power system were brought together to build a power budget for the proposed launch pad. By considering all of the electrical equipment and the maximum load expected on the associated wiring, the apparent power of the field was deduced. Using SysML, the electrical loads of end devices are understood quickly by parametric links. Based on this information, the electrical engineers may determine wire gauge, conduit or cable tray sizing, and feel confident that the foundation being poured will support the long-term goals of the mission. A final design problem involving the selection of hydraulic cylinders was performed. This exercise exhibited how a mechanical engineer may go about sizing a hydraulic lift circuit capable of meeting lifting requirements. Future hydraulic design will include specific selection of actuators, flow rates, pump sizing, and model integration.

The MBSE model for a small class spaceport provides an early vision of a project's future trajectory for stakeholders. Design options are better understood when traceability and system relationships are comprehended early on. The systems are then subjected to trade studies and analysis, providing project participants with the information necessary to determine the preferred design. By capturing the critical technical information within a robust model, a range of mission needs can be supported. A modeling language, clear MBSE method, and system modeling tool are the key elements enabling the proposed approach. The modeling architecture described offers a starting point for implementing the ISO/IEC 15288 technical framework with SysML as the medium. By bringing together competent engineers across multiple disciplines, the essential knowledge of each domain may be better represented for complex systems in one model.

BIBLIOGRAPHY

1. "AFT Arrow | Fluid Dynamic Simulation Tool." *Www.Aft.com*, www.aft.com/products/arrow. Accessed 24 Nov. 2020.
2. Alaska Aerospace Corporation, *A Diversified, Sustainable Aerospace Industry for Alaska* (2011), 7.
3. American National Standards Institute, and National Fire Protection Association. *Standard for Purged and Pressurized Enclosures for Electrical Equipment in Hazardous Locations.* Boston, Ma, National Fire Protection Association, 2020.
4. Amissah, Matthew. "A Framework for Executable Systems Modeling." Engineering Management & Systems Engineering Theses & Dissertations, 1 July 2018, digitalcommons.odu.edu/emse_etds/31/, 10.25777/f1h6-e712. Accessed 30 Oct. 2020.
5. Andress, Jason, and Steve Winterfeld. Cyber Warfare : Techniques, Tactics and Tools for Security Practitioners. Amsterdam ; Boston, Elsevier, 2014.
6. "Application Engineering Guide Cylinder Engineering Aids Cylinder Modifications Ordering Information." Miller Fluid Power. Available at: https://www.parker.com/literature/Industrial%20Cylinder/cylinder/cat/english/M9905.pdf
7. Beery, Paul T.. "A Model-Based Systems Engineering Methodology for Employing Architecture In System Analysis: Developing Simulation Models Using Systems Modeling Language Products to Link Architecture and Analysis." (2016).
8. Cardno, Ph.D, Catherine A. "New Launch Pad Designed for Smaller Rockets | ASCE." Www.Asce.Org, 4 Aug. 2015, www.asce.org/magazine/20150804-new-launch-pad-designed-for-smaller-rockets/. Accessed 30 Oct. 2020.
9. *Cardoso, E.*, Study of the Critical Point of Several Difficultly Liquifiable Gases: Nitrogen, Carbon Monoxide, Oxygen and Methane, *J. Chim. Phys. Phys.-Chim. Biol.*, 1915, 13, 312.
10. "Class/Division Hazardous Location" Rockwell Automation, Allen Bradley. Available at: *https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/800-wp003_-en-p.pdf*
11. CONTROL VALVE HANDBOOK Fifth Edition. Emerson Automation Solutions, Sept. 2019.
12. Crane Co. Engineering Division. Flow of Fluids through Valves, Fittings and Pipe. Stamford, Ct, Crane Co, 2013.
13. "Cryogenic Liquids." *Ehs.Princeton.Edu*, ehs.princeton.edu/book/export/html/184#:~:text=Vaporization%20%2D%20All%20cryogenic%20liquids%20produce. Accessed 24 Nov. 2020.
14. "Darcy-Weisbach Pressure and Major Head Loss Equation." *Engineeringtoolbox.com*, 2019, www.engineeringtoolbox.com/darcy-weisbach-equation-d_646.html.
15. DoD. 1998: "DoD Modeling and Simulation (M&S) Glossary" in DoD Manual 5000.59-M. Arlington, VA, USA: USA: US Department of Defense. January. P2.13.22. Available at: *http://www.dtic.mil/whs/directives/corres/pdf/500059m.pdf*

16. Fernandez, Jose and Hernández Corbato, Carlos. (2019). Practical Model-Based Systems Engineering.
17. Foust, Jeff. "Firefly Aerospace to Take over Vandenberg Delta 2 Pad." *SpaceNews*, 2 May 2018, spacenews.com/firefly-aerospace-to-take-over-vandenberg-delta-2-pad/. Accessed 24 Nov. 2020.
18. Friedenthal, S. and Oster, C., 2017. Architecting Spacecraft With Sysml. CreateSpace Independent Publishing Platform.
19. Hutchinson Jr., Virgil. A Model-Based Systems Engineering Approach to the Heavy Lift Launch System Architecture Study. Orbital ATK Space Systems Group, Dulles, VA
20. ISO/IEC 15288:2008, Systems and Software Engineering – System Life Cycle Processes. Int Organ Standardization/Int Electrotechnical Comm March 18, 2008.
21. ISO/IEC 19514:2017, Information technology-Object management group systems modeling language (OMG SysML). Int Organ Standardization/Int Electrotechnical Comm May 2017
22. Janene Scully/Associate Editor. "Report: Falcon Plan OK for Environment." *Santa Maria Times*, 5 Feb. 2011, santamariatimes.com/news/local/military/vandenberg/article_9fb48ac2-31be-11e0-890e-001cc4c002e0.html. Accessed 24 Nov. 2020.
23. Kaslow, David & Anderson, Louise & Asundi, Sharan & Ayres, Bradley & Iwata, Curtis & Shiotani, Bungo & Thompson, Robert. (2015). Developing a CubeSat Model-Based System Engineering (MBSE) Reference Model-interim status. 10.1109/AERO.2015.7118965.
24. Kovacic, Michael, et al. An Illustrated Guide to Electrical Safety. Des Plaines, Ill., American Society Of Safety Engineers, 2011.
25. Kriening, Torsten. "Sea Launch to Be Restored at Cost of about $470 Million." *SpaceWatch.Global*, 28 Aug. 2020, spacewatch.global/2020/08/sea-launch-to-be-restored-at-cost-of-about-470-mln/. Accessed 24 Nov. 2020.
26. "Liquid Oxygen." *www.ehs.ufl.edu*, www.ehs.ufl.edu/programs/lab/cryogens/oxygen/. Accessed 24 Nov. 2020.
27. London, Brian. (2012). A Model-Based Systems Engineering framework for concept development.
28. "Measures of Effectiveness (MOE) - AcqNotes." *AcqNotes*, 20 June 2018, acqnotes.com/acqnote/tasks/measures-of-effectivenessrequirements#:~:text=Measure%20of%20Suitability%20(MOS)&text=MOSs%20typically%20relate%20to%20readiness. Accessed 24 Nov. 2020.
29. Model-based Systems Engineering: An Integrated Approach – An Ovum white paper for Dassault Systemes, February 2012, Tony Baer, Michael Azoff
30. NASA. "NASA - NSSDCA - Spacecraft - Details." *Nasa.Gov*, 2019,nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1957-001B.
31. NASA. "NASA - SpaceX Successfully Launches Falcon 1 to Orbit." *Www.Nasa.Gov*, 28 Sept. 2008, www.nasa.gov/offices/c3po/home/spacex_falcon1_flight_4.html. Accessed 13 Dec. 2020.
32. NASA Requirements for Ground-Based Pressure Vessels and Pressurized Systems (PVS), NASA-STD-8719.17 Revision C
33. NASA. 2007. Systems Engineering Handbook, Revision 1. Washington, DC, USA: National Aeronautics and Space Administration (NASA). NASA/SP-2007-6105.

34. Nelson, B. and Dr. J. Simmons. Rocket Flight Termination Behaviors Bridged to Analytical Simulation in MBSE, Orbital ATK/FSG/LVD and Phoenix Integration
35. "NIST Chemistry WebBook - SRD 69." Nist.Gov, National Institute of Standards and Technology, 19 Aug. 1997, webbook.nist.gov/chemistry/.
36. Object Management Group. OMG Systems Modeling Language ™ (SysML®). V1.5. Available at: *http://www.omg.org/spec/SysML/*.
37. Osipov, Viatcheslav & Daigle, Matthew & Muratov, Bahovuddin & Foygel, Michael & Smelyanskiy, V. & Watson, Michael. (2011). Dynamical Model of Rocket Propellant Loading with Liquid Hydrogen. Journal of Spacecraft and Rockets. 48. 987-998. 10.2514/1.52587.
38. Patterson, Trina. "Pegasus Rocket." *Northrop Grumman*, www.northropgrumman.com/space/pegasus-rocket/. Accessed 24 Nov. 2020.
39. Pearce, P. and M. Hause. "ISO-15288, OOSEM and Model-Based Submarine Design." (2012).
40. Rhodes, Donna & Ross, Adam. (2010). Anticipatory Capacity: Leveraging Model-Based Approaches to Design Systems for Dynamic Futures.
41. Roberts, Thomas G. "Spaceports of the World." Aerospace Security, Center for Strategic and International Studies, Mar. 2019, aerospace.csis.org/data/spaceports-of-the-world/. Accessed 30 Oct. 2020.
42. "Rocket Lab Increases Electron Payload Capacity, Enabling Interplanetary Missions and Reusability." *Rocket Lab*, 4 Aug. 2020, www.rocketlabusa.com/about-us/updates/rocket-lab-increases-electron-payload-capacity-enabling-interplanetary-missions-and-reusability/. Accessed 24 Nov. 2020.
43. Sanford Friedenthal, Alan Moore, and Rick Steiner. 2008. A Practical Guide to SysML: Systems Modeling Language. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
44. Space-Track.org, accessed November 24, 2020, https://www.space-track.org/.
45. "SysML FAQ: What Is an Internal Block Diagram (IBD)?" *SysML.org*, sysml.org/sysml-faq/what-is-internal-block-diagram.html. Accessed 24 Nov. 2020.
46. "SysML Specifications - Current Version: OMG SysML 1.6." *SysML.org*, sysml.org/sysml-specs/. Accessed 24 Nov. 2020.
47. "Systems Modeling Language (SysML) | Enterprise Architect User Guide." *Sparxsystems.com*,Sparxsystems.com/enterprise_architect_user_guide/15.2/model_domains/sysml.html. Accessed 24 Nov. 2020.
48. Virgin Orbit and Virgin Galactic Web Sites, LauncherOne Updates and Launch Vehicle Description, 2015-17. LauncherOne Service Guide Version 0.2, Virgin Galactic, March 25, 2016.
49. Walden, David D, et al. Systems Engineering Handbook : A Guide for System Life Cycle Processes and Activities. Hoboken, New Jersey, Wiley, 2015.
50. Weilkiens, Tim. "MBSE Tools." *Model Based Systems Engineering*, mbse4u.com/sysml-tools/.
51. Wright, David, et al. The Physics of Space Security : A Reference Manual. Cambridge, Ma, American Academy Of Arts And Sciences, 2005.

VITA

Walter McGee Taraila is a Master of Science student studying Aerospace Engineering in the Department of Mechanical and Aerospace Engineering at Old Dominion University (238 Kaufman Hall, Norfolk, VA 23529). In 2012, he received his Bachelor of Science in Aerospace Engineering from the University of Maryland, College Park. He is currently employed as the Chief Engineer for Launch Pad 0C supporting venture class launches at the Mid-Atlantic Regional Spaceport (MARS) in Wallops Island, Virginia. He has eight years of experience working on the Fluid Control System for Launch Pad 0A and has sat on console for the previous nine Antares launches supporting Northrop Grumman in their efforts to resupply the International Space Station. His research interests include data science, computer science, mathematical finance, cryptography, flow control, and space systems.