2022

Model-Based Systems Engineering for a Small-Lift Launch Facility

Walter McGee Taraila

Sharanabasaweshwara Asundi
Old Dominion University

Original Publication Citation
Taraila, W. M., & Asundi, S. (2022). Model-based systems engineering for a small-lift launch facility. *Systems Engineering*, *25*(6), 537-550. https://doi.org/10.1002/sys.21634

Received: 13 March 2021 | Revised: 9 May 2022 | Accepted: 13 June 2022

DOI: 10.1002/sys.21634

REGULAR ARTICLE

WILEY

Model-based systems engineering for a small-lift launch facility

Walter McGee Taraila[1] | Sharanabasaweshwara Asundi[2]

[1]Design Engineering, Mid-Atlantic Regional Spaceport, Norfolk, Virginia, USA

[2]Department of Mechanical and Aerospace Engineering, Old Dominion University, Norfolk, Virginia, USA

Correspondence
Walter McGee Taraila, Mid-Atlantic Regional Spaceport, Norfolk, VA, USA.
Email: wtara001@odu.edu

Title Page: The following manuscript adheres to the ethics and integrity policies implemented for The International Council on Systems Engineering (INCOSE) Journal. The authors received no financial support for the research, authorship, or publication of this manuscript. The authors grant John Wiley & Sons, Inc. permission to reproduce material from other sources as required. The data that support the findings of this study are available from the corresponding author upon reasonable request.

Abstract

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 ground systems on a launch pad servicing a small class payload (0–2 tons). With the projected growth in launch cadence of small-lift rockets in the coming decade, there is a need to design increasingly complex launch systems with greater efficiency. The potential improvements in project communication, quality, and productivity are explored by developing a model following the ISO/IEC 15288 technical process framework and the International Council on Systems Engineering (INCOSE) Object-Oriented Systems Engineering Method (OOSEM) methodology. The stakeholder requirements are defined and analyzed to provide traceability to individual systems and subsystems. An architecture is proposed by generating engineering artifacts such as piping and instrumentation drawings. The concepts are verified and validated by performing engineering trade studies concentrated on the pneumatic and fuel subsystems.

KEYWORDS
aerospace engineering, launch complex, launch pad, MBSE, model-based systems engineering, small-lift launch facility, spaceport, SysML, systems engineering

1 | INTRODUCTION

The National Aeronautics and Space Administration (NASA) defines systems engineering as "a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system".[1] Model-based systems engineering (MBSE) is an approach where the design, analysis, specifications, and verification details are captured within a model representing the system.[2,3] The blueprint is controlled throughout the project's lifecycle to offer stakeholders a single source of truth with traceability to the mission requirements. The utilization of a system model can offer significant benefits over the document-based historical approach including a reduction in error propagation, improved maintainability of the system-of-interest, reusability, and a heightened understanding. Graphical modeling languages enable this application, with Systems Modeling Language (SysML) as the de-facto standard that was adapted and adopted by the Object Management Group (OMG).[4] Within a SysML model, elements are abstracted offering overviews of components, interfaces, constraints, and interconnections between systems.

A launch pad is an example of a systems engineering project in which the primary mission is to provide all the resources required to launch a rocket vertically from the ground. The facility includes a launch mount to physically support the rocket, service structures for umbilical mates, and all the infrastructure that the launch service provider (LSP) requires prior to liftoff. For a liquid-propellant rocket, ground systems typically include civil, mechanical, controls, fluid ground support, cryogenics, propellants, deluge, pneumatics, HVAC, safety, environmental, electrical power systems, and hydraulics. The design and maintenance of a launch pad is a complex engineering problem. The systems must function cohesively in parallel to support the payload reaching the targeted orbit. Early in the development of a launch pad, life cycle costs tend to become locked into place. Late identification and repair to problems cost considerably more than problems caught earlier. Descopes to mission requirements accepted later versus earlier in the project life cycle also result in reduced cost savings. Figure 1 illustrates NASA's launch propulsion systems roadmap.[5] To support the expected growth in launch cadence of small-lift vehicles, there is a need to apply MBSE to ground based spaceports.

Each launch facility is custom built and tailored to meet the needs of the LSP. Whether the customer is a commercial company 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. To get to market at the fastest pace with a quality result, spaceports require well-organized modeling tools with a single source of truth. There are currently limited resources for applying MBSE to launch facilities, which is the primary motivation for this research. The application of an MBSE methodology and technical framework to the lifecycle processes of a launch pad offers the aerospace practitioner a state-of-the-art case study contributing methodologically to the field of systems engineering.

One of the first steps to the MBSE approach is to select the appropriate methodology and framework. An MBSE methodology is defined as a collection of processes, methods, and tools to aid systems engineering within a "model-driven" context.[2,6,7] It is essentially a "recipe", applying tools, methods, and processes to a specific problem set. The leading MBSE methodologies used in industry today include IBM Telelogic Harmony-SE,[8] International Council on Systems Engineering (INCOSE) Object-Oriented Systems Engineering Method (OOSEM),[9,10,11] IBM Rational Unified Process for Systems Engineering for Model-Driven Systems Development,[12] Vitech MBSE Methodology,[13] JPL State Analysis,[14,15] and Dori Object-Process Methodology.[16] An in-depth survey of each methodology is beyond the scope of this report, but the reader is directed to Jeff Estefan's survey of MBSE methodologies for a detailed description of those listed above.[6]

This paper is organized as follows. Section 2 describes the technical process framework and methodology selected to develop the proposed launch facility. The subsequent sections are driven by the framework, beginning in Section 3 with the definition and analysis of stakeholder requirements. Sections 4–7 describe the architectural design, implementation, integration, verification, and later stages of the project. This is where the engineering solutions that satisfy the requirements are explored, studied, realized, iterated upon, and improved. Section 8 discusses the overall process and lessons learned. Finally, Section 9 presents the conclusions of the study.

SIGNIFICANCE AND PRACTITIONER POINTS

- The paper aligns with the vision of INCOSE, "A better world through a systems approach", by applying model-based systems engineering to launch pad design, a unique use case, which has minimal documentation for researchers or practitioners. The small-class launch industry is projected to follow an exponential growth curve over the next decade. To support the trend, there is a significant need to design increasingly complex launch systems with an improved rate of efficiency. The venture class launch service providers are obtaining licenses from the Federal Aviation Administration, which allow hundreds of missions per year. In this context, the MBSE breakdown of the launch pad schematics, flow analyses, etc., would be of critical interest to a researcher in the field of aerospace engineering. For practitioners in the rocket launching community, the application of INCOSE's OOSEM methodology, and the ISO/IEC 15288 technical process framework will aid in the development of future requirements, pad architectures, test plans, and construction of launch pads while using model-based approaches in SysML.

2 | INCOSE OOSEM AND ISO/IEC 15288

INCOSE OOSEM, developed in 1998, is often implemented in conjunction with SysML to provide a best practice to MBSE.[17,18] The present-day model-based methodologies are dominated by the object-oriented approach.[19,20] The term "object-oriented" (OO) is derived from the third generation of software programming languages, succeeding assembly and machine-code.[21] A higher level of abstraction

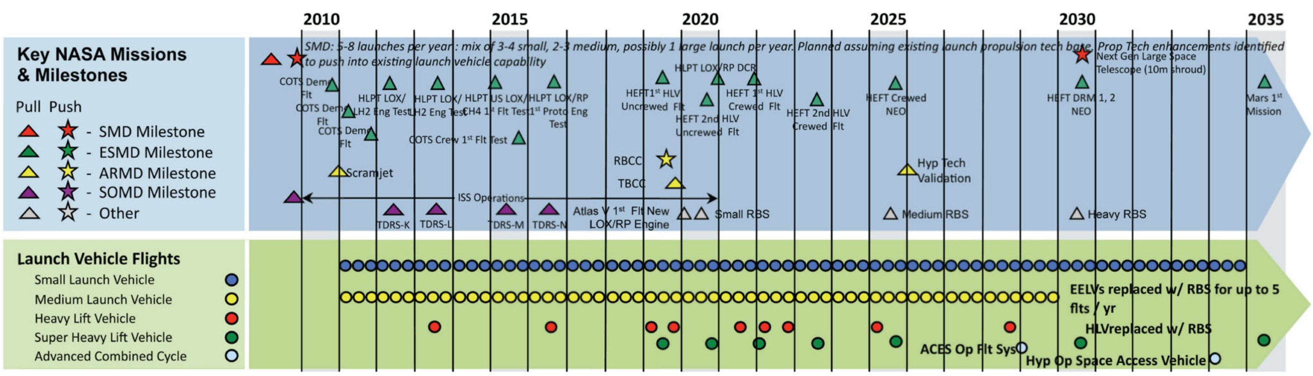

FIGURE 1 NASA launch propulsion systems roadmap[5]

is obtained in OO with the introduction of classes, objects, aggregation, and inheritance. OOSEM outlines a methodology founded on model-based and object-oriented techniques while following traditional system engineering practices.[17] The processes are iterative and recursive in nature, matching the intentions of the ISO/IEC 15288 framework.[22] 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. OOSEM development activities include analyzing stakeholder needs, defining system requirements, defining logical architectures, synthesizing candidate allocated architectures, optimization, evaluation of alternatives, and finally system validation and verification. 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 research paper.[23]

The ISO/IEC 15288 technical process is the framework selected to capture a launch pad design with MBSE and SysML as the medium. ISO/IEC 15288 is an international standard managed by ISO/IEC JTC 1/SC 7 concerning systems and software engineering processes and lifecycle stages.[22,24] The standard is divided into the four primary categories of Technical, Project, Agreement, and Enterprise. The technical processes include stakeholder requirements definition, requirement analysis, architectural design, implementation, integration, verification, transition, validation, operation, maintenance, and disposal.[25] The ISO/IEC 15288 technical process was selected because the steps are complimentary to the OOSEM process.[22] It was also chosen because INCOSE has adopted the 15288 standard and has integrated some elements into the INCOSE Systems Engineering Handbook v3.[26]

3 | STAKEHOLDER REQUIREMENTS DEFINITION AND ANALYSIS

The first step of the ISO/IEC 15288 technical process is to produce a list of stakeholder needs regarding the system-of-interest.[22,24] The inputs include a description of the objectives, timeline, budget, constraints, terms, conditions, and industry standards or specifications. The output is a formally documented and accepted set of requirements to govern the project.[25] The traditional text-based requirements are bridged to the modeling environment using the medium of SysML.

In this application, requirements have been organized into a branching hierarchy classified by level, type, and system. The requirement levels are mission (Level 1, ID-#), system (Level 2, ID-#.#), and subsystem (Level 3, ID-#.#.#) and the requirement types are functional, performance, constraint, or verification. The systems include site, controls, fluid ground support equipment (FGSE), safety, electrical power systems (EPS), and hydraulics. The functional requirements define what an element must do, and the performance requirements quantify how well an element must accomplish a specific function. The constraint requirements capture the regulatory, safety, environmental, or operational limitations, as well as design standards enforced. The verification requirements dictate methods of establishing confidence to ensure the system will perform as intended in its environment. The system requirement decomposition is exemplified in Figure 2 concentrating on the FGSE system text-based requirements that will be linked to the model. FGSE requirements are guided by the NASA requirements for ground-based pressure vessels and pressurized systems.[27]

To manage requirements and the relationships between them, requirement management tools are commonly used and maintained in a database. To bridge text-based requirements to the MBSE model, SysML offers requirement modeling functionality which works in parallel with a well-defined project requirement management process to enable rigorous traceability. For example, the requirements in the FGSE specification are modeled in a tree structure consistent with the organizational structure of the text-based specification. To perform requirement analysis in SysML, relationships are defined that link requirements to other requirements and model elements. These requirement relationships include derivation, satisfaction, verification, refinement, and trace. Once the text-based requirements are loaded into the model, they can function in an object-oriented machine-readable manner paving the way for eventual verification test cases.[17]

4 | ARCHITECTURAL DESIGN

The next step in the ISO/IEC 15288 technical process is "Architectural Design", aligning with the OOSEM technical process step to "Define Logical Architecture" and "Synthesize Allocated Architectures" to fulfill stakeholder requirements.[22,24] The inputs include the requirements, design constraints, a traceability matrix, and system interface specifications. The outputs are an architectural design baseline, system element descriptions, refined interface requirements, and an initial verification strategy.[25] To produce these outputs, the model is set up with a package structure, modeling conventions are defined, a mission analysis is outlined, and internal block diagrams (IBDs) are generated to represent traceability to requirements and external interfaces such as the launch vehicle.[28]

The system IBD (Level 2) in Figure 3 shows connection points between launch pad systems as well as external mission elements. The controls, FGSE, safety, site, launch vehicle, EPS, and range ground station are all parts of the mission. The ports on each subsystem boundary identify each of its external interfaces. Each port represents unique interfaces such as safety, electrical, tubing and piping, software, or more general abstract logical connection points. The frame corresponds to the launch pad and interfaces with the launch vehicle are labeled with "LV i/f". For example, the "LV fuel i/f" port specifies the external interface to the launch vehicle's first stage fuel tank and is the point where FGSE performance requirement ID-3.4.2 for fuel flow rates between 0 and 100 gpm are verified. From an object-oriented perspective, a launch vehicle interface can be viewed as an abstraction; a definition of the general properties of an FGSE mechanical interface that it shares with other interfaces, such as electrical umbilicals or network connections. Furthermore, the FGSE mechanical interface "inherits" the general properties of a launch vehicle interface (e.g., port

FGSE Requirements

ID-#	Type	Description
	Mission Requirements	
ID-1	Launch Cadence (Level 1)	The launch pad shall support a mission every two months
ID-2	**Functional Requirements**	
ID-2.2	**FGSE Functional** (Level 2)	
ID-2.2.1	FGSE Detank (Level 3)	The launch pad shall detank all commodities in the event of a launch scrub
ID-2.2.2	FGSE Loading	The launch pad shall load all commodities from storage area(s) to launch vehicle
ID-2.2.3	FGSE System I/F	The FGSE subsystems shall provide interface conditions [ID-3.4.1 through 3.4.9]
ID-2.2.4	FGSE Sample Frequency	The FGSE subsystems interfacing with launch vehicle shall be sampled 1 month prior to each mission
ID-2.2.5	FGSE System Cleanliness	The FGSE subsystems shall meet cleanliness requirements at each interface
ID-3	**Performance Requirements**	
ID-3.4	**FGSE Performance**	
ID-3.4.1	Fuel Subsystem - Storage	The Fuel Subsystem shall contain 30,000 gallons
ID-3.4.2	Fuel Subsystem - Flow	The Fuel Subsystem shall provide flowrates between 0 to 100 gpm
ID-3.4.3	Fuel Subsystem - Pressure	The Fuel Subsystem shall operate with a Maxiumum Expected Operating Pressure (MEOP) of 150 psig
ID-3.4.4	Fuel Subsystem - Filtration	The Fuel Subsystem shall contain inline filtration of 10 µm
ID-3.4.5	Fuel Subsystem - Purge	The Fuel Subsystem shall have a Gaseous Nitrogen purge.
ID-3.4.6	Helium Subsystem - Storage	The Helium Subsystem shall contain 100,000 standard cubic feet (scf)
ID-3.4.7	Helium Subsystem - Flow	The Helium Subsystem shall provide flowrates between 0 to 15 standard cubic feet per minute (scfm)
ID-3.4.8	Helium Subsystem - Pressure	The Helium Subsystem shall operate with a Maximum Expected Operating Pressure (MEOP) of 5,000 psig
ID-3.4.9	Helium Subsystem - Filtration	The Helium Subsystem shall contain inline filtration of 10 µm
ID-4	**Constraints**	
ID-4.2	**FGSE Constraints**	
ID-4.2.1	FGSE P&ID	The FGSE system shall document Piping and Instrumentation (P&ID) drawings for all commodities
ID-4.2.2	FGSE Cleanliness	The FGSE components shall meet SAE AS4059 Rev E. Class 2 cleanliness standard
ID-4.2.3	FGSE PVS Standard	The FGSE system shall follow standard NASA STD 8719.17 NASA Requirements for pressure systems
ID-4.2.4	FGSE Temperature	The FGSE system shall operate within an ambient temperature range of 0ºF to 120ºF
ID-4.2.5	FGSE Pressure Vessel Code	The FGSE pressure vessels shall conform to ASME B&PV Section VIII Division 1 standard
ID-5	**Verification**	
ID-5.2	**FGSE Verification**	
ID-5.2.1	FGSE Isometrics	The FGSE system shall document isometrics with dimensions of all components and assemblies
ID-5.2.2	FGSE Leak Tests	The FGSE system shall contain documentation for initial leak checks of all subsystems
ID-5.2.3	FGSE Fabrication	The FGSE system shall document component certifications of conformance
ID-5.2.4	FGSE System P&ID	The FGSE system shall document functional diagrams of end components with interconnections
ID-5.2.5	FGSE System Codes	The FGSE system shall conform to standard pressure vessel and pressure system standards
ID-5.2.6	FGSE Inspection Plan	The FGSE system shall contain documentation of inspection plans for each commodity
ID-5.2.7	FGSE Weld Procedures	The FGSE system shall document B31.3 compliant weld procedures
ID-5.2.8	FGSE Hydrostatic Testing	The FGSE system shall document all hydrostatic testing reports
ID-5.2.9	FGSE Mechanical Checkout	The FGSE system shall document mechanical checkouts
ID-5.2.10	FGSE Non-Destructive Exam	The FGSE system shall document all non-destructive testing
ID-5.2.11	FGSE Relief Valve Analysis	The FGSE system shall document relief valve analysis for all subsystems
ID-5.2.12	FGSE Flow Analysis	The FGSE system shall generate a flow analysis for the proposed design of each subsystem

FIGURE 2 SysML requirement table for launch pad FGSE

diameter, material, location) and then defines additional FGSE mechanical interface-specific properties (e.g., fitting type, maximum expected operating pressure, torque, thread diameter), which can then be inherited by other variants of FGSE mechanical interfaces.[23] The model has the ability to tie a complex system of systems together in an object-oriented fashion, assisting in the generation of derived artifacts such as 3D CAD models, power distribution plans, network mapping, piping and instrumentation diagrams (P&IDs), functional diagrams, site plans, and hazardous classification maps. For additional guidelines on modeling methods and conventions, the reader is directed to "A Practical Guide to SysML"[17] by Friedenthal, Moore, and Steiner. For supplementary applications of MBSE models, the reader is pointed to "Architecting Spacecraft with SysML",[28] by Sanford Friedenthal and Christopher Oster.

In this application, the launch pad is further defined by a subsystem (Level 3) decomposition. The control subsystem provides the ability to remotely run hazardous operations, establish situational awareness for operators, offer hazardous event detection, and capture data acquisition. The architecture proposed includes primary and redundant synchronized servers. There is a fiber optic connection to an off-site location for remote operation. The electrical vault houses the servers, network switches, and fiber patch panels, which tie to field controllers for the fuel storage, oxidizer storage, and launch mount areas. Every controller has multiple network connections, increasing redundancy by offering two independent paths back to the electrical vault. By generating the architecture in SysML, single point failures are quickly identified and eliminated. The control system is a critical component for mission success, which provides remote telemetry to operators, automates events, and helps to lift, lower, load, de-tank, and monitor the rocket prior to lift-off.

The field controllers accept telemetry from end devices for valve position, pressure, temperature, flow, level, and other signal types by primarily using analog or digital inputs. Controllers also send commands to end devices using analog or digital outputs. With the control system IBD and fundamental architecture established, the responsible engineer can select hardware, generate control drawings, and

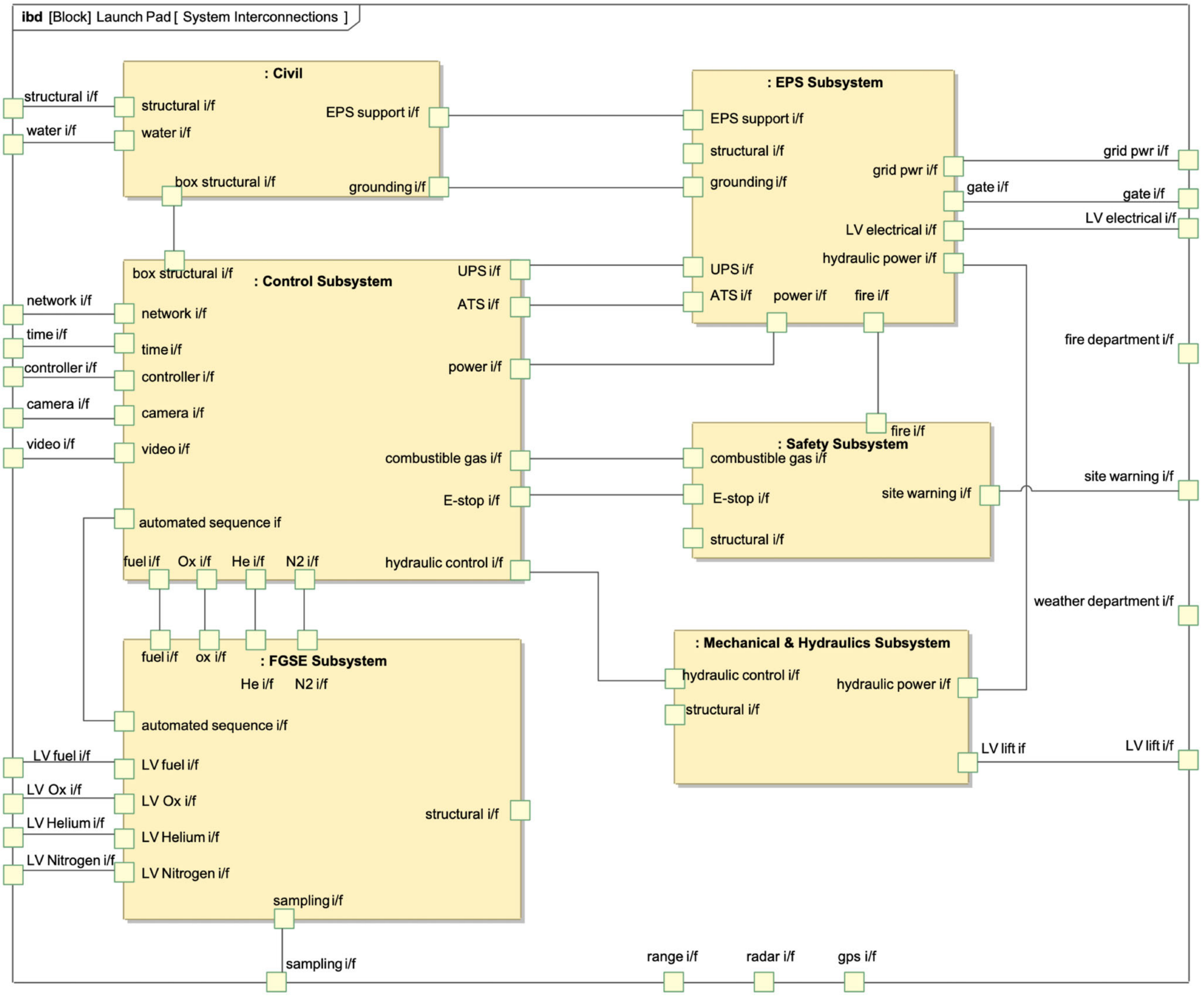

FIGURE 3 Launch pad IBD of system interfaces (Level 2)

begin the implementation process. From a safety perspective, the hardware selected shall be explosion-proof, intrinsically safe, purged and pressurized,[29] or another hazardous area mitigation technique outlined in NFPA 70 Article 500.7.[30] The Level 3 control architecture IBD is shown in Figure 4.

5 | IMPLEMENTATION

After requirements are developed and an architectural design is generated, the subsequent step is to design and fabricate each system to conform to the architecture. The 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.[25] In this study, the implementation focuses on the generation of P&ID diagrams for the FGSE helium subsystem.

The FGSE helium subsystem performance requirements (ID-3.4.6 through ID-3.4.9) guide the design of the P&ID functional diagram shown in Figure 5. The launch pad is required to store 100,000 standard cubic feet of supply, provide flow rates at each interface between 0 and 15 standard cubic feet per minute (scfm), handle a maximum expected operating pressure (MEOP) of 5000 pounds per square inch (psi), and contain 10 micrometer inline filtration. The system is intended to purge the liquid oxygen subsystem and launch vehicle. The P&ID is not a SysML artifact, but is based on the requirement tables and IBD interfaces defined in the previous figures. Every component in the diagram has a unique identifier. For example, PT-H36 is a pressure transducer in the helium subsystem measuring supply pressure to the LO_2 P&ID with a unique tag denoting where the interface is located on the corresponding LO_2 drawing. The launch pad helium subsystem evolves with traceability by generating a requirements table, deriving P&IDs based on the requirements, extracting a bill of materials (BOM) from the P&ID, and linking the associated blocks within

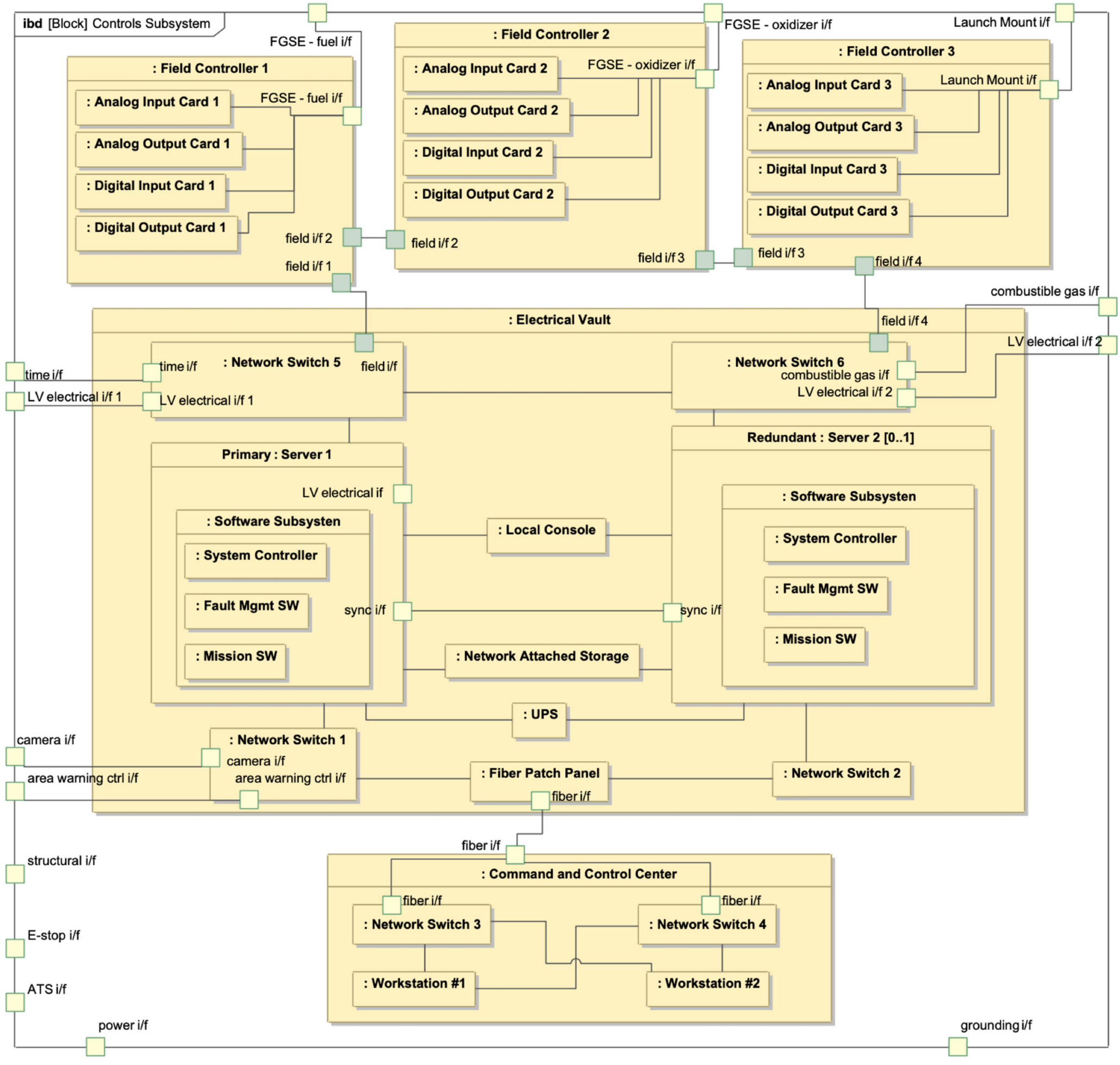

FIGURE 4 Control subsystems IBD (Level 3)

the SysML model to their respective requirements. If a unique identifier has no link to a requirement, it serves no purpose and should be removed.

The methods for verification of the helium subsystem will use this P&ID architecture as the baseline. A similar P&ID and verification strategy is deployed for other commodities on the launch pad. In SysML, every component in the helium P&ID may be represented as a block with attributes that would traditionally be stored inside a BOM. The valve flow coefficients, pressure tranducer ranges, hazardous mitigation techniques, component dimensions, relief valve set pressures, and sensor accuracies are examples of possible block attributes stored in the SysML environment.

6 | INTEGRATION AND VERIFICATION

Integration realizes the system by gradually combining elements of the architectural design. This is an iterative process repeated in parallel with the verification and validation processes. The outputs are a verifiable system, the results of the integration testing, and records of problem resolution as needed. During integration, components are received, installed, and records are archived for future reference. The verification process seeks to determine if the system was built correctly. Inputs of verification are the baseline system requirements, test criteria, a requirement verification traceability matrix, and system elements to be verified. The activities include defining a strategy

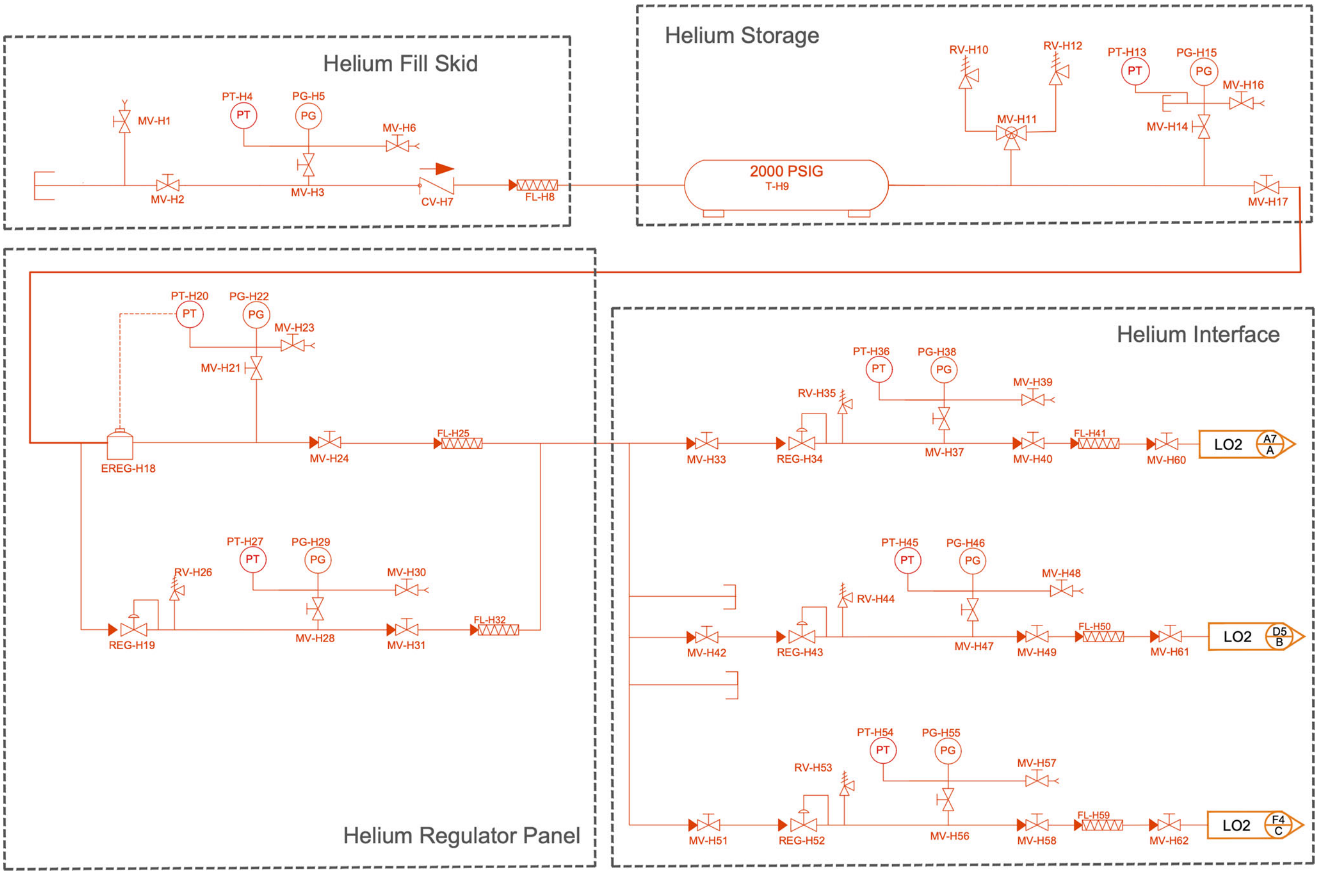

FIGURE 5 P&ID for FGSE helium subsystem

for system verification, creating, and maintaining a requirements verification traceability matrix, and conducting specific verifications to demonstrate compliance with requirements.[25] The four fundamental methods of verification are inspection, testing, analysis, and demonstration. There are numerous verification activities to perform for a launch pad. The verification process is modeled by building a set of requirements, generating diagrams based on the requirements, which are used to build a set of schematics standardized in the industry. Once the schematics are developed, these are translated to calculations, the results of which trace back to the initial requirement with a clear pass or fail criteria. The following section conducts a verification by analysis with respect to the proposed FGSE helium subsystem architecture.

6.1 | Gaseous helium verification trade study

The goal of this exercise is to verify the helium specification (ID-3.4.7) by performing an analysis by hand and comparing the results to a flow analysis simulation. 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. 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.

The helium simulation results are generated by using a recognized commercial-off-the-shelf flow analysis software package for compressible flow titled *AFT Arrow*.[31] In this example, the "length marching" approach is implemented wherein the panel piping is divided into many 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 with analytical calculations which are accurate at low velocities, and thus low Mach numbers, providing an independent second verification prior to building the system. The calculation methodology starts with a desired mass flow rate and volumetric flow rate, which is provided by the LSP and used to calculate the expected pressure drop and flow velocity. Calculations shown in Table 1 are based on the Crane "*Flow of Fluids*" Handbook.[32] The AFT Arrow flow analysis model shown in Figure 6 is built to represent the P&ID diagram shown in Figure 5 for the gaseous helium (GHe) subsystem.

The flow starts with junction, J1, representing a storage tank currently holding a pressure of 2000 psig. The piping is ½ inch diameter throughout the entire model. There are two pressure reducing regulators in parallel, J5 and J34, which join at a tee directly downstream. For

TABLE 1 Analytical calculations for the helium subsystem[32]

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 °F
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	3311.81
M	Mach number	N/A	0.0305
Re	Reynolds number	N/A	4432.24
ΔP	Pressure loss	$psig$	2.0592

Inputs denoted with an asterisk (*).

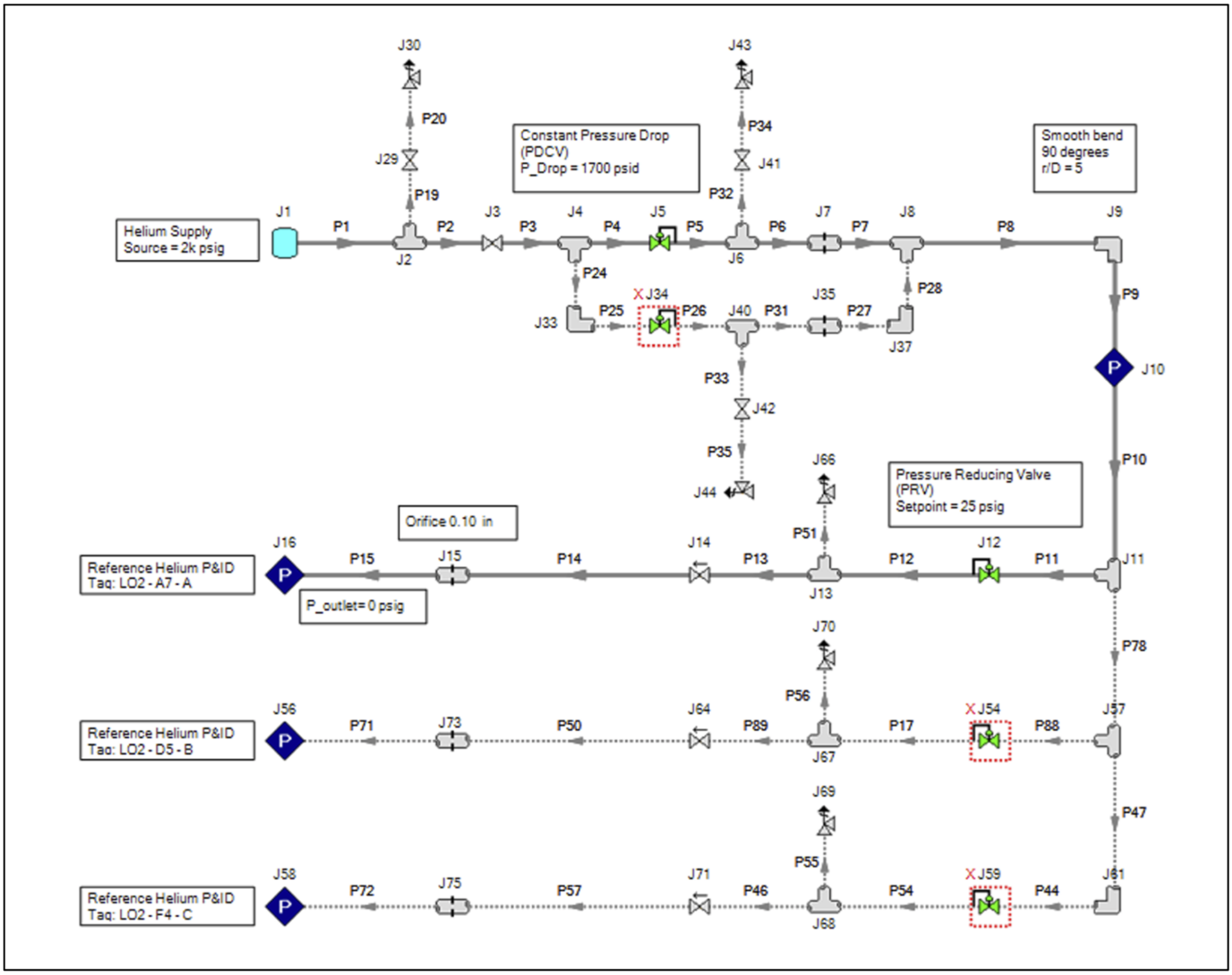

FIGURE 6 AFT arrow FGSE helium fluid dynamic model configuration

Pipes | Axial Points | Heat Transfer

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 7 AFT arrow FGSE helium fluid dynamic model output

the initial simulation, the J34 regulator is simulated fully closed. Only one of the three supply lines is modeled in this example. 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. The outlet #1, J16, has a 0.10-inch orifice to deliver a volumetric flow rate of 14.8841 scfm. The 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 outputs depicted in Figure 7 and the analytical results shown in Table 1 verify requirement ID-3.4.7. The flow analysis is completed prior to system construction. Confidence that the proposed design will meet stakeholder needs is increased with the verification of the two independent analyses. Figure 8 demonstrates the conversion of the analytical calculations to a parametric diagram (PAR) within the SysML model for reusability and refinement. The output from the gaseous helium trade study is plugged into the parametric diagram which is traced to requirement ID-3.4.7 and linked within the model.

6.2 | Fuel verification trade study

The fuel subsystem requirements specify 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. To meet these requirements, the fuel subsystem P&ID is simplified and represented by Figure 9.[33] Bernoulli's principle (Equation 1) uses 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.[32]

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

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 (Equation 2) 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 resistance coefficients for piping, elbows, ball valves, and bends are summed together to find K_{Total} (Equation 3), which is an input to calculate the final head loss expected at the launch mount interface (Equation 4). The fuel density, height change, inlet pressure, and head loss are inputs to calculate the interface pressure at Point #4 (Equation 5).

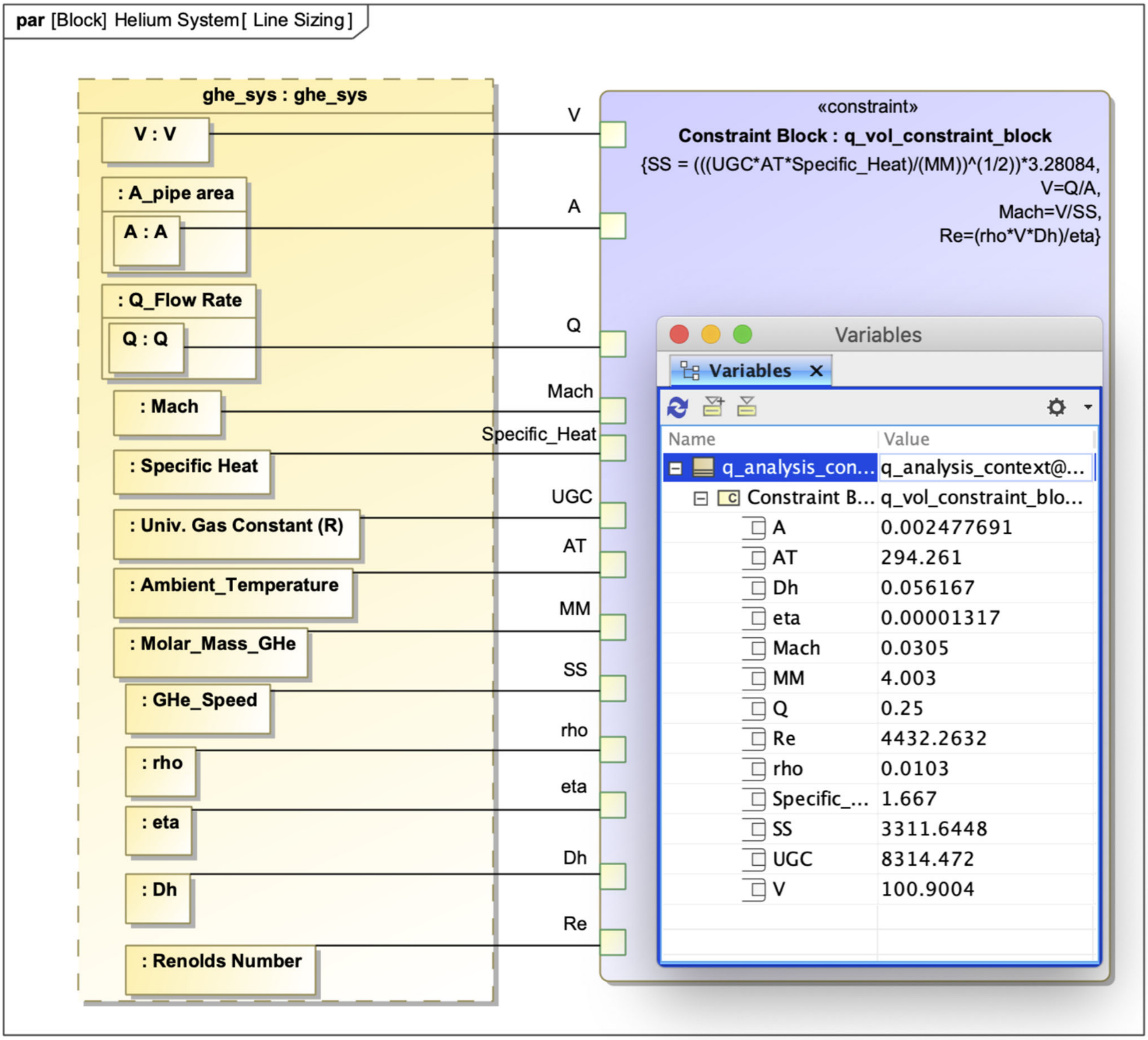

FIGURE 8 Parametric Diagram (PAR) for helium verification

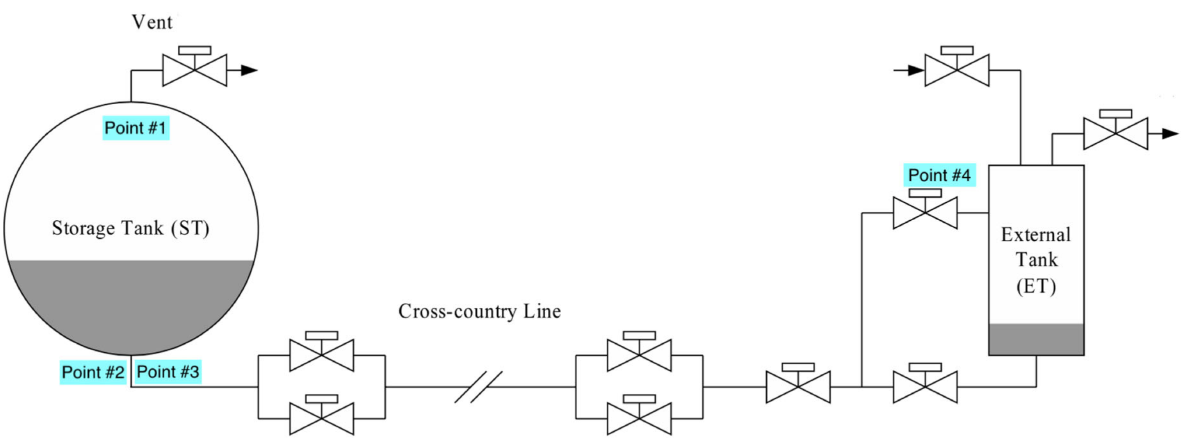

FIGURE 9 Fuel analysis model setup[33]

TABLE 2 Analytical calculations for the fuel subsystem[32]

P1	Pressure point #1 – Top of tank	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

$$R_e = \frac{Q\rho}{d\mu} \tag{2}$$

$$K_{Total} = \sum_{i=1}^{n} K_i \tag{3}$$

$$h_L = 0.00259\frac{Q^2 K_{Total}}{d^4} \tag{4}$$

$$P_{outlet(Point\#4)} = \rho\left(H_3 - H_4 + \frac{P_3}{\rho} - h_L\right) \tag{5}$$

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. The following Table 2 summarizes the inputs and calculated outputs of the fueling subsystem analysis model. FGSE fuel performance requirement for flow rate (ID-3.4.2) is verified by analysis, with an estimated volumetric flow rate of 103.34 gpm. The fuel analysis is guided by equations and examples listed in the Crane technical paper[32] and Emerson's Control Valve Handbook.[34]

7 | LATER STAGES OF OOSEM AND ISO/IEC 15288 TECHNICAL FRAMEWORK

The process of transition entails the transfer of custody and responsibility from the development team to the operational and support organizations. Following a successful transition, the validation process confirms that the realized system complies with the defined requirements of the mission. Validation of the system is subject to the approval of the key stakeholders. The operational process is the successor to validation, using 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. The maintenance process helps to sustain the system and extend the lifetime to support ongoing operations. In this manuscript, Cameo Systems Modeler, an industry leading cross-platform MBSE environment is deployed to develop a launch pad model. The software provides robust tools to define, track,

FIGURE 10 Launch pad 39C at NASA Kennedy Space Center[36]

and visualize all aspects of a complex system-of-systems in compliant SysML models and diagrams.[35] After the launch pad is constructed, the next steps are mechanical checkout, performance testing, Wet Dress Rehearsal, and finally the launch to orbit. A launch pad is an example of a batch process plant where the majority of the system's operational life period is spent in an operation, maintenance, and optimization mode preparing for the next mission.

A novel and methodological advancement to the state of the art in MBSE can be applied to the later stages of the framework in the form of pad optimization. As the SysML model is frozen and configuration is locked down for the first launch, it is possible to export the object-oriented software model to file formats such as UML XMI 2.4, MagicDraw Native XML, EMF Ecore, MOF XMI, or Eclipse UML2.[35] Upon export, software engineers can use this file in the object-oriented programming language of their preference, such as Python, C++, Ruby, or C#, to provide stakeholders with added benefits in the form of automation. A key metric for determining whether a launch is successful or not is by performing a post-launch data analysis. The system, modeled as a set of objects which are controlled and manipulated in a modular manner, can be represented as a set of classes, dictionaries, and functions. These software templates have already been refined and linked via the SysML model, making post launch data analysis more efficient. Test cases to validate the specifications and requirements are streamlined with the object-oriented approach. Useable implementations of the UML metamodel may be used post-launch in support of further development of the modeling tools. Lessons learned from the first launch are realized in a more efficient manner, reducing the time required to optimize the batch process plant's capabilities in support of the next mission. The SysML model is updated after the first launch with configuration changes selected by the stakeholders who use the post-launch data reports to determine the best path forward. The SysML model is finalized, frozen once again, and the configuration is locked down for the second launch. This iterative process, aligned with ISO/IEC 15288 and OOSEM, efficiently executes each mission until no longer required. The final step of the technical framework is disposal, removing the system from the operational environment with the intent of concluding its use.[25]

8 | DISCUSSION

A study of MBSE applied to a small-lift launch facility was presented using tools such as SysML and AFT Arrow, the INCOSE OOSEM methodology, and the ISO/IEC 15288 technical process framework. The development and analysis of stakeholder requirements in an object-oriented MBSE environment helped to shape the project's direction, focusing on the constraints in a prioritized manner. The stakeholder requirements definition and analysis phase were driven by objectives, timeline, budget, etc., and resulted in a set of structured requirements. These requirements along with design constraints, interface specifications, etc., were fed as inputs to the architectural design phase. The architectural design created a system solution that fulfilled the requirements, which was depicted in a system level IBD diagram showcasing all the launch pad systems as well as a subsystem level IBD connecting elements of the proposed control system. The IBDs, a critical outcome of the architectural design phase, along with verification and validation criteria, industry standards, and safety practices, were used as inputs to the implementation phase. The overarching goal of the implementation phase was demonstrated in the form of P&ID diagrams. For this study, P&ID diagrams for the FGSE, which offered the engineer a crucial end-to-end system graphic with the intent to expedite later stages of the framework, were developed. As part of the integration and verification phase, the FGSE trade studies were performed for the gaseous helium and fuel subsystems to increase confidence in the design prior to development. For completeness, a plan of action in the form of potential activities (validation, operations, maintenance, and disposal) was identified for the remainder of the OOSEM and ISO/IEC 15288 technical framework. The next steps are to complete a second iteration, refine the MBSE model, perform additional trade studies, improve the requirements, and build the post-launch object-oriented software to support future missions. To give the reader a better understanding of the product, Figure 10 depicts an artist's representation of NASA Kennedy Space Center Launch Pad 39C[36] designed for small-lift launch vehicles.

9 | CONCLUSION

As small-lift LSPs ambitiously race to capture market share in the exponentially growing small class payload market, launch pads are required to support this trend.[37] A spaceport is an example of a complex systems engineering application which has a need for the advantages offered by MBSE. The cost of the project and the associated timeline to completion are locked into place at early life cycle stages, compelling launch pad engineers to draft elegant and accurate solutions during initial iterations. By following the ISO/IEC 15288 technical framework, INCOSE OOSEM methodology, and applying SysML to the MBSE approach, a proof of concept for expediting launch pad development is realized.

CONFLICTS OF INTEREST

The authors have no conflicts of interest to disclose.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

1. NASA. *Systems Engineering Handbook, Revision 2*. National Aeronautics and Space Administration (NASA); 2016. NASA/SP-2016-6105.
2. Friedenthal S, Griego R, Sampson M, NCOSE model based systems engineering (MBSE) initiative. In: INCOSE 2007 symposium, vol. 11. 2007.
3. Beery PT, 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.
4. Amissah M. *A Framework for executable systems modeling*. Engineering Management & Systems Engineering Theses & Dissertations; 2018. https://doi.org/10.25777/f1h6–e712. Accessed: 30 Oct. 2020.
5. Mcconnaughey P, et al. DRAFT Launch Propulsion Systems Roadmap Technology Area 01. 2010.
6. Estefan JA. *Survey of Model-Based Systems Engineering (MBSE) Methodologies, Rev. B, INCOSE Technical Publication, Document No.: INCOSE-TD-2007-003-01*. International Council on Systems Engineering; 2008.
7. Hart LE, Introduction to Model-Based System Engineering (MBSE) and SysML. Presented at the Delaware Valley INCOSE Chapter Meeting; 2015
8. Douglass BP. *The Harmony Process, I-Logix White Paper*. I-Logix, Inc.; 2005.
9. Friedenthal S. *Object Oriented Systems Engineering. Process Integration for 2000 and Beyond: Systems Engineering and Software Symposium*. Lockheed Martin Corporation; 1998.
10. Lykins H, Friedenthal S, Meilich A, Adapting UML for an Object-Oriented Systems Engineering Method (OOSEM). Proceedings of the INCOSE 2000 International Symposium, Minneapolis, MN; 2000.
11. Object Oriented System Engineering Method. *OOSEM Descriptive Outline for INCOSE SE Handbook Version 3, Annotated Update, Sect. 6.4.2*. Object Oriented System Engineering Method; pp. 1–6. 2006.
12. Kruchten P. *The Rational Unified Process: An Introduction*. 3rd ed. Addison-Wesley Professional; 2003.
13. Childers SR, Long JE. *A Concurrent Methodology for the System Engineering Design Process, Vitech White Paper*. Vitech Corporation; 2004.
14. Ingham M. *State Analysis Overview: What PSEs Ought to know, Briefing Slides (Internal Document), Jet Propulsion Laboratory*. California Institute of Technology; 2006.
15. Rasmussen B. *Session 1: Overview of State Analysis, (Internal Document), State Analysis Lite Course, Jet Propulsion Laboratory*. California Institute of Technology; 2005.
16. Dori D. *Object-Process Methodology: A Holistic Systems Paradigm*. Springer-Verlag; 2002.
17. Friedenthal S, Moore A, Steiner R. *A Practical Guide to SysML: Systems Modeling Language*. Morgan Kaufmann Publishers Inc.; 2008.
18. Walden DD, Roedler GJ, Forsberg K, Douglas Hamelin R, Shortell TM. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. John Wiley & Sons; 2015.
19. Sage AP, Rouse WB. *Handbook of Systems Engineering and Management*. John Wiley & Sons; 2014.
20. Madni AM, Sievers M. Model-based systems engineering: motivation, current status, and research opportunities. *Syst Eng*. 2018;21(3):172-190.
21. Kiran GA, Haripriya S, Jalote P, Effect of object orientation on maintainability of software. In: *1997 Proceedings International Conference on Software Maintenance*, pp. 114-121. IEEE; 1997.
22. International Organization for Standardization (ISO). *ISO/IEC/IEEE 15288: 2015–Systems and Software Engineering–System Life Cycle Processes*. International Organization for Standardization (ISO); 2015.
23. Pearce P, Hause M, ISO-15288, OOSEM and Model-Based Submarine Design. 2012.
24. *ISO/IEC 15288:2008, Systems and Software Engineering – System Life Cycle Processes*. Int Organ Standardization/Int Electrotechnical Comm; 2008.
25. *INCOSE Systems Engineering Handbook, A Guide for System Life Cycle Processes and Activities, INCOSE-TP-2003-002-03.2*. International Council on Systems Engineering; 2010.
26. Haskins C, ed. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, v. 3, INCOSE-TP-2003-002-03*. International Council on Systems Engineering; 2006.
27. NASA Requirements for Ground-Based Pressure Vessels and Pressurized Systems (PVS), NASA-STD-8719.17 Revision C.

28. Friedenthal S, Oster C. *Architecting Spacecraft with Sysml*. CreateSpace Independent Publishing Platform; 2017.
29. American National Standards Institute, and National Fire Protection Association. *Standard for Purged and Pressurized Enclosures for Electrical Equipment in Hazardous Locations*. National Fire Protection Association; 2020.
30. *Class/Division Hazardous Location*. Rockwell Automation, Allen Bradley. Available at: https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/800-wp003_-en-p.pdf
31. AFT Arrow | Fluid Dynamic Simulation Tool. Www.Aft.com, www.aft.com/products/arrow Accessed: 24 Nov. 2020
32. Crane Co. *Engineering Division. Flow of Fluids through Valves, Fittings and Pipe*. Crane Co; 2013.
33. Osipov V, Daigle M, Muratov B, Foygel M, Smelyanskiy V, Watson M. Dynamical model of rocket propellant loading with liquid hydrogen. *J Spacecr Rockets*. 2011;48:987-998.
34. *Control Valve Handbook*, 5th ed. Emerson Automation Solutions; 2019.
35. Cameo Systems Modeler - CATIA - Dassault Systèmes®.
36. *KSC Launch Pad 39C*. The Planetary Society. www.planetary.org/space-images/ksc-launch-pad-39c Accessed: 24 Dec. 2021
37. Roberts TG. *Spaceports of the World*. Aerospace Security, Center for Strategic and International Studies; 2019. aerospace.csis.org/data/spaceports-of-the-world/. Accessed: 30 Oct. 2020.

AUTHOR BIOGRAPHIES

Walter McGee Taraila P.E. is a Senior Engineer at the Mid-Atlantic Regional Spaceport (MARS) located in Wallops Island, Virginia. He is the Chief Engineer for a Venture Class launch facility with ten years of experience working on ground control systems. He has sat on console as the Fluid Control System lead for MARS during Antares launches from Pad-0A to resupply the International Space Station (ISS). He received his Bachelor of Science in aerospace engineering from the University of Maryland, College Park in 2012 and a Master of Science in aerospace engineering from Old Dominion University in 2020. He is a member of the International Society of Automation (ISA) and the American Institute of Aeronautics and Astronautics (AIAA).

Sharanabasaweshwara Asundi, Ph.D., an Assistant Professor in the Department of Mechanical and Aerospace Engineering at Old Dominion University (ODU), is an expert in space systems engineering and has engaged in the design, development of several space systems, facilities, including an amateur radio ground station and two magnet coil test facilities. He is a Federal Communications Commission licensed amateur radio operator. He is engaged in several teaching and research activities, largely focused on furthering the Space Systems Engineering Program at ODU. He has engaged in research collaboration with National Aeronautics and Space Administration as a Science Collaborator and a Principal Investigator and has been awarded grants by the U.S. Air Force Office of Scientific Research, U.S. Department of Agriculture, and National Science Foundation, among others.

How to cite this article: Taraila W, Asundi S. Model-based systems engineering for a small-lift launch facility. *Systems Engineering*. 2022;25:537–550.
https://doi.org/10.1002/sys.21634