Integrating Model Based Systems and Digital Engineering for Crewed Mars Mission Planning

Mitchell Kirshner,[1] and Dr. Ricardo Valerdi.[2]
The University of Arizona, Tucson, Arizona, 85705, USA

With technological advances in the 21st century and the rise of the commercial space industry, crewed Mars missions to land humans on our red neighbor have become increasingly feasible. One of the key challenges in accomplishing this goal is determining whether Model Based Systems Engineering training is necessary for space systems engineers to adopt Digital Engineering. This paper reviews Model Based Systems Engineering benefits and limitations, as well as historically significant crewed Mars missions that planned to land on the surface. We present models for a Mars Transit Habitat developed in the modeling language SysML using Cameo Systems Modeler. These models leverage Cameo's simulation capabilities to execute MATLAB scripts integrating Digital Engineering software Systems Toolkit into the modeling process, demonstrating a reusable interoperability methodology for Mars mission planning not yet existing in this domain's literature. We conclude that integrating Model Based Systems Engineering with Digital Engineering for crewed Mars landing mission planning can lead to multiple benefits for space systems Digital Engineering, including (1) greater system simulation capabilities, (2) enhanced model fidelity, and (3) improved system understanding. Future research to further enhance these benefits includes adding cybersecurity considerations and greater system details to the models.

[1] PhD Student, Department of Systems and Industrial Engineering.
[2] Professor, Department of Systems and Industrial Engineering.

I. Introduction

The growth of Systems Engineering in the 21st century has brought about new techniques to design, plan, test, and implement space systems. Many of these techniques have developed into specialized fields, including Model Based Systems Engineering (MBSE) and Digital Engineering (DE). Together, MBSE and DE enable a system-centric approach to architecture design that can help engineers produce better systems with lower cost and risk. Now more than ever, humanity's ability to settle nearby terrestrial objects is a reality; colonies on Mars are increasingly possible. However, a key challenge remains in the education of systems engineers who will develop crewed Mars missions: do space engineers need to learn MBSE to successfully adopt DE?

To answer that question, we explore the integration of MBSE and DE in mission architecture planning for facilitating crewed Mars surface exploration. We shall leverage existing MBSE architecture design methods and historical Mars missions to create models implemented through both MBSE and DE software together as a proof-of-concept methodology to aid space engineers in mission development and analysis. In doing so, we provide evidence for the following benefits to crewed Mars landing mission planning arising from the combination of MBSE and DE: greater system simulation capabilities, enhanced model fidelity, and improved system understanding.

II. Literature Review

A. MBSE Strengths and Opportunities

While engineers can use textual specifications to develop missions in DE environments, MBSE holds the potential to greatly improve on that process. Compared to textual specifications, modeling can more effectively create, manage, and verify data while promising consistency across engineering activities through the use of MBSE languages such as Systems Modeling Language (SysML) and Unified Modelling Language (UML) [1]. SysML has distinguishing features that make it superior to UML in our application, which we shall evaluate in later sections [2]:

1. SysML allows descriptions of numerical quantities through standardized ISO units.
2. SysML has robust semantics detailed in formal specifications.
3. SysML models can form a coherent system body of knowledge that interfaces with external tools [2].

SysML's strong graphical semantics aid in visually depicting a system. However, SysML is weak in executable semantics for models of computation, timing and time synchronization, and large-scale event representations [3]. Despite such limitations, SysML still has the potential to aid in crewed Mars landing mission planning.

B. MBSE Space Systems Modeling Limitations

Achieving an optimal configuration for space habitats and groups of habitats is a complicated system engineering problem with many configurations and trade space considerations. MBSE provides space engineers using DE a suitable approach to characterize architectures of the various systems in the problem space and define scenarios that affect those systems. However, MBSE methodologies lack some capabilities, including the following:

1. Native interoperability between Digital Engineering tools and systems. The synthesis of Digital Engineering and MBSE is new in literature and has yet to become a common practice in either field.
2. Literature that describes the use of MBSE techniques often does not provide empirical evidence of its purported benefits. Henderson and Salado empirically observe this discrepancy in their 2021 paper [4].
3. Evaluation of an architecture's cybersecurity. MBSE techniques do not inherently measure and compare threat models of each architecture against known goals for security.

The methodology we shall demonstrate, which benefits space systems digital engineers who use MBSE, intends to address each of these shortcomings; we shall elaborated upon this in a later section.

C. MBSE for Crewed Mars Mission Planning

There has been no holistic MBSE development for a top-level architecture of a crewed Mars mission; while NASA has developed SysML models for some systems of these missions including integrated power and avionics systems and deep space habitats [5], no studies to date have explored the integration of these models with each other to form a system of systems model, let alone integration of external physics-based DE environments. That being said, NASA has conducted research in recent years on developing methodologies to handle complicated mission plans using MBSE and Model Based Mission Assurance [6]; NASA has yet to fully implement these methodologies.

NASA directorates are still creating new top-level descriptions of full-scale missions without leveraging MBSE. The NASA Human Exploration and Operations (HEO) Mission Directorate updated their HEO Exploration Objectives document to include Mars missions in the 2030s [7]. The NASA Mars Study Capability Team has been developing a Deep Space Transport and Mars Mission Architecture that comprises phases 3 and 4 of the HEO baseline [8]. Meanwhile, the NASA Langley Space Technology Mission Directorate has used models for their Mars Architecture Technology Drivers Overview without mentioning MBSE [9].

D. Cybersecurity

For future space colonization, cybersecurity is more than just a factor to consider; even with MBSE and DE software handling system complexity, space engineers must constantly consider cybersecurity throughout the entire systems development process. Modern space systems are cyber-physical, meaning they depend on the synergy of computational and physical components [10]. As such, a breach in cybersecurity could prove fatal to a mission if for example, a hacker impeded navigations causing a vehicle to crash. Furthermore, malicious agents seeking valuable data could take advantage of unencrypted instrument data onboard the space system [10].

Space systems literature as recent as 2018 discuss the many challenges associated with cybersecurity, as well as the lack of support infrastructure to improve space asset security like space-specific standards or information sharing organizations [11]. This is true of both government and commercial space systems alike; fortunately recent studies have expounded the need to mitigate space cyber threats and listed potential attack types [12]. Unfortunately, cybersecurity principles for space systems are not standardized in neither the public nor private sectors, as researchers have only proposed them in the last 3 years [13]. While our research does not propose a methodology for capturing these cybersecurity principles in mission plan models, space engineers must be aware of their importance.

III. MBSE Application to Crewed Mars Mission Planning

Scientists and engineers have worked towards Martian colonization for decades. While historically government agencies have dominated Martian system design development, modern growth of the privatized space sector has resulted in a rise of commercial involvement. To evaluate the extent to which developers have implemented MBSE in crewed Mars landing mission planning in both the public and private sector, we review past and ongoing plans.

A. Martian Colonization: Noteworthy Plans and Designs

In 2016, the NASA Engineering Safety Center began the MBSE Pathfinder effort to develop and advance MBSE capabilities across NASA, apply MBSE to real issues, and capture issues and opportunities surrounding MBSE [14]. Over 20 development programs at NASA Jet Propulsion Lab (JPL) have applied MBSE, including the Mars 2020 mission and the Jupiter Europa Orbiter [15]. JPL uses MBSE to explore more comprehensive options for space systems and to perform validation of system designs with a reduction of paper management for the design engineers, thereby improving the quality of communications between system and subsystem engineers [15]. Table 1 describes historical crewed Mars surface missions that planned to land on Mars.

Table 1: Summary of notable proposed crewed missions to Mars.

Name	Year	Goals	Source
Von Braun Das MarsProjekt	1952	Set ten 4,000-ton ships with 70 crewmembers in Earth orbit using 950 ferry flights to settle the Martian poles. Failed to adequately account for radiation exposure.	[16]
Lewis Research Center Mars Mission Profile	1957	Land seven crew members on Mars with a piloted Mars Landing Vehicle; 420-day round trip with a 40 day stay at Mars in May 1971. Precursor to NASA's formation.	[16]
Mars Excursion Module (MEM)	1963	After 120 days, crew members would move from the main spacecraft to the NASA MEM landing vehicle for surface operations.	[16]
North American Rockwell MEM	1967	Updated 1963 MEM to incorporate Mariner 4 results and minimize cost. Changed diameter to 30 feet and shape like the Apollo Command Module. Six landing legs would enable it to set down on a 15-degree slope.	[16]
SAIC Mars Mission Design	1984	Deliver over 160 metric tons of spacecraft components to Earth orbit with 18 launches. A four-person crew would travel to Mars in a spacecraft with sub-vehicles including a conical, two-stage lander to be abandoned on Mars post-mission. Cost estimated at then-year $38.5 billion.	[16]
NPO Energia's Mars Landing Vehicle	1991	Launch two astronauts vertically and land horizontally for surface living space. After a week, the cosmonauts would ascend to the orbiter and return to Earth. This Soviet nuclear Mars mission differed from previous conical designs: it was cylindrical, with a conical forward section that fit the Energia rocket's payload envelope.	[16]
Mars Direct	1991	Aerobrake a piloted lander into Mars orbit. Crew would spend 500 days on Mars. Made use of established concepts: split mission architecture, an artificial gravity tether, and a conjunction-class mission plan.	[16]
NASA Mars Design Reference Mission (DRM) 3	1997	Utilize an Earth-Return Vehicle reliant on In-situ Resource Utilization (ISRU) for six astronauts. Based on the Lewis Research Center Mars Mission Profile. "Scrubbed" compared to the 1993 version: reduced estimated mission cost by minimizing spacecraft weight.	[16]
Boeing Mars Transfer Vehicle and Lander Concepts	2006	8-year cycle of missions between 2031-2038. Multiple vehicles were proposed, including a cargo lander with a reusable ISRU habitat and a crewed ascent-descent vehicle. Example of a modern commercial company developing their own Mars landing mission architecture.	[17]
NASA Design Reference Architecture 5.0	2009	Updated and most recent version of the NASA Mars DRM. Described in detail in later sections.	[18]
SpaceX Starship	2019	Carry roughly 100 people to the moon or Mars with 6 main phases: launch and booster return, ship orbit arrival, ship refuel via tankers that return to Earth, travel to Mars, refuel on Mars with ISRU, and Mars ascension/direct return to Earth. Commercial company.	[19] [20]

The mission plans in Table 1 never explicitly mention usage of MBSE or SysML. Companies like SpaceX, Blue Origin, Virgin Galactic, and others may use SysML for developing their platforms, but might not share that information. The NASA Design Reference Architecture (DRA) 5.0 comes closest to utilizing MBSE as it defines a systems architecture by which to evaluate mission plans [18]. The DRA 5.0 contains systems common to many plans in Table 1, especially the 1957 Mars Mission Profile and the 1997 Mars Design Reference Mission 3 [16].

B. Models of Habitat and Infrastructure Systems

The taxonomy in Figure 1 summarizes breakdown of the physical systems described in the DRA 5.0. While the DRA 5.0 explicitly defines many subsystems and components complete with values for predicted mass, it generalizes others such as the Heavy-Lift Launch Vehicle; this example specifically outsourced the specification of such a vehicle to using the now-cancelled Ares 5 rocket [18]. Since the last update to the DRA 5.0 in 2009, however, SpaceX rockets such as Starship have become increasingly capable of supporting the role of Heavy-Lift Launch Vehicle, with the added feature of reusability. Modeling endeavors to create an executable SysML diagram of a Mars mission should consider creating hybrid variations of the DRA 5.0 to include new technologies like Starship. Furthermore, surface systems modelling should cover cybersecurity considerations.

Because NASA last officially updated the DRA 5.0f in 2009, it is ostensibly obsolete for modern Mars planning. While creating an extensible architecture, which accounts for design alternatives like Starship, requires additional research, NASA should postulate a new reference architecture in the future leveraging MBSE. This architecture should incorporate plans across all of NASA's directorates and integrate preexisting models of mission subsystems [7]–[9]. NASA has developed architectures for crewed Mars missions that integrate design alternatives to form one comprehensive architecture for the expedition; a new DRA must account for such integrations [21]. We hope that our research in MBSE and DE integration for crewed Mars landing mission planning will benefit future NASA architecture development in order to contribute to the overall mission of landing humans on Mars.

C. Modeling the Design Reference Architecture with SysML

SysML software Cameo Systems Modeler can create an MBSE product based on Figure 1. Not only can one model the high-level system architecture, but one can also create a Block Definition Diagram (BDD) for each subsystem. We elected to limit the scope of our modeling endeavor to a subsection of the DRA 5.0, and chose to strictly adhere to NASA's architecture design choices. The DRA 5.0 describes several systems in detail, including the Mars Transit

Habitat, comprising: the power system, avionics, extravehicular activity (EVA) systems, the structure, environment control and life support, thermal management systems, crew accommodations, and food [18]. Figure 2 shows these systems in the smaller diagram. While the DRA 5.0 does not explicitly mention how the architecture considers water for crew members in the transit habitat, it mentions using ISRU for Surface Systems to create drinking water.

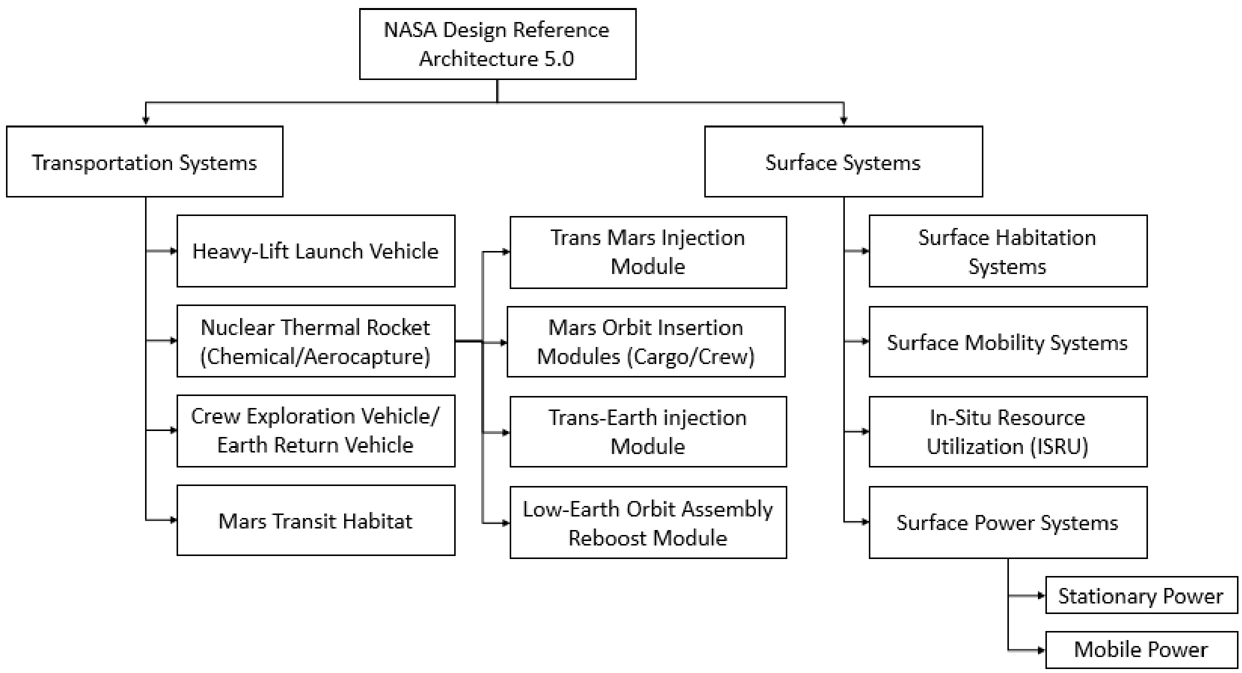

Figure 1: Summary of the DRA 5.0's breakdown of required systems for a crewed Mars mission [18].

We developed the Mars Transit Habitat model on the bottom of Figure 2 using values from the DRA 5.0 to include details about mass and stowed volume of applicable components [18]. We chose the Mars Transit Habitat because of its detail in the DRA 5.0 and literature precedent of SysML modeling [22]. To utilize SysML's internal abilities to add the masses of components and pass those value to the parent object in the model hierarchy, parametric diagrams linked the various component values using constraint blocks that formulaically added values when executed using Cameo's simulation software. Figure 3 shows an Instance BDD with computed values.

The calculated total mass of the transit habitat is equal to the values given by the DRA 5.0: 41,340kg [18]. However, the total stowed volume differs from the sum of all the added stowed volume values provided in the DRA 5.0, showing a discrepancy in the DRA 5.0. We chose mass as the value to model due to its ease of understanding and inclusion in the DRA 5.0. Although adding mass values may seem trivial computationally, researchers can apply this calculation methodology to much more complex equations through properly defined parametric diagrams.

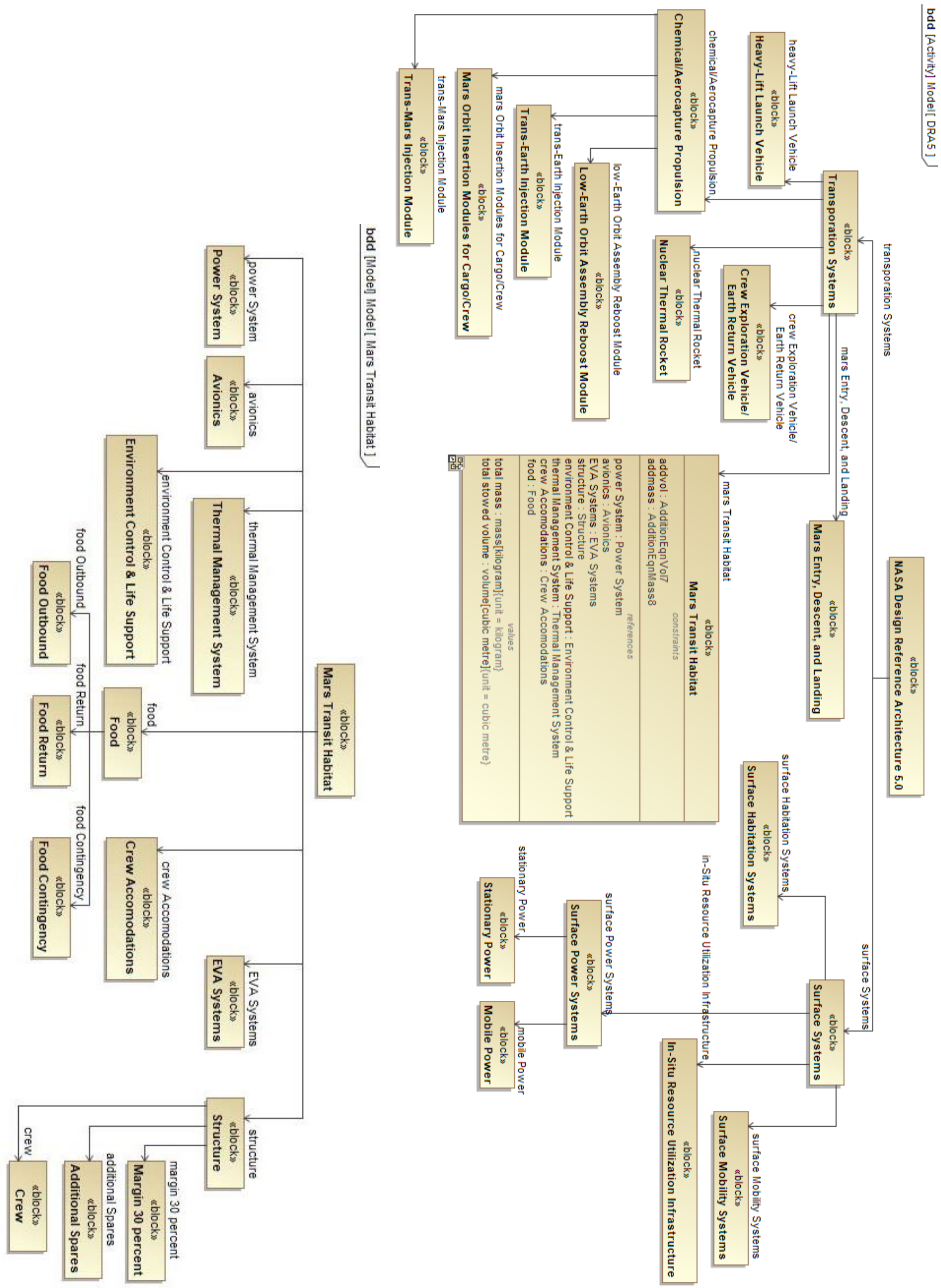

Figure 2: The NASA Design Reference Architecture 5.0 depicted in SysML alongside a detailed Block Definition Diagram (BDD) of the Mars Transit Habitat, a Transportation Subsystem.

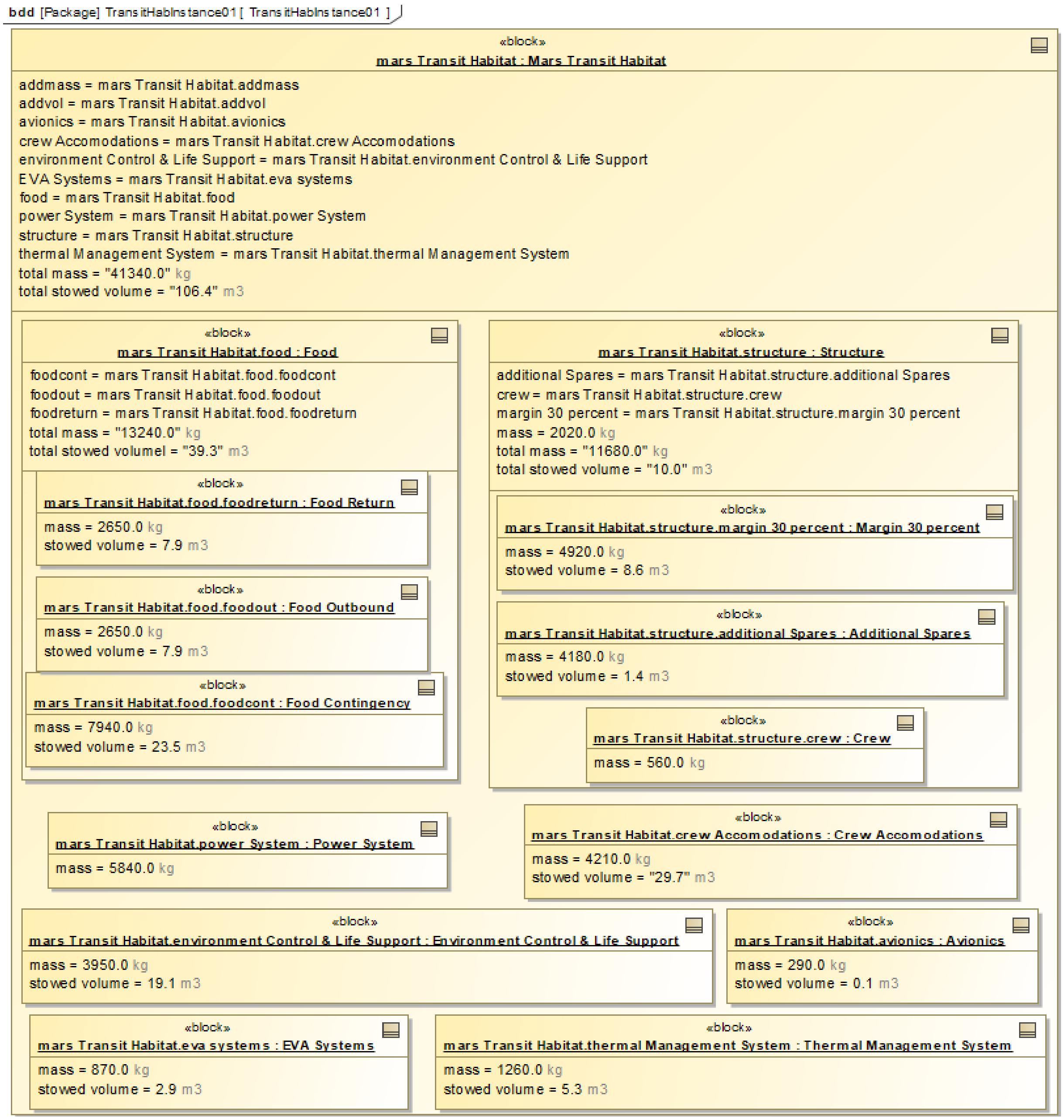

Figure 3: Instance of the DRA 5.0 Mars Transit Habitat BDD with summed values.

IV. MBSE Integration with Digital Engineering

Because of Cameo's ability to interpret MATLAB commands, one can effectively link models to Analytical Graphic Inc.'s (AGI's) System Toolkit (STK) to create an interoperable physics-based simulation environment that executes and tests SysML models. STK contains features to digitally engineer space mission architectures and analyze space environment conditions that may be too complex for architects to constantly consider. AGI regularly hosts events in which experts and leaders in the DE industry share capabilities at the forefront of the field [23].

A. MATLAB Scripts for STK Automation with BDDs

Cameo's constraint blocks and parametric diagram features extend beyond adding mass values: can also reference MATLAB scripts and functions in a specified directory. Parametric diagrams can pass values to MATLAB functions defined in the constraint block used for that diagram and then pass the results of that function back into the Cameo model environment. On the DE side, MATLAB has multiple ways to connect to STK: MexConnect, which communicates over TCP/IP, as well as COM connection [24]. Thus, one can use MATLAB code to achieve Cameo MBSE interoperability with AGI DE software STK. AGI plans to release software for STK that will permit direct SysML integration without MATLAB; AGI has not released that software at the time of writing [25].

When executed, the MATLAB function in the Appendix opens STK and places an object with an input mass in orbit around Mars. Cameo implements the MATLAB function as a BDD constraint block in the parametric diagram shown in Figure 4. The parametric diagram allows the SysML models to pass the total mass value from the Mars Transit Habitat to STK. STK simulates the system in a digital environment and returns a value for the sample orbit's semi-major axis. Use of STK's DE environment affords 3D model visualization, in-depth analyses, and now MBSE software interoperability with a DE physics engine. Because MBSE can handle values passed from STK and vice versa, there is low risk in this integration procedure of possible information loss, as semantics between the modeling environments can interface through parametric diagrams like Figure 4. While typically SysML might struggle with time-critical events and other specialized tasks due to its weak semantics, SysML allows for non-normative constructs with MATLAB integration, making it possible to extend SysML beyond a visual representation tool and leverage languages with stronger semantics [26].

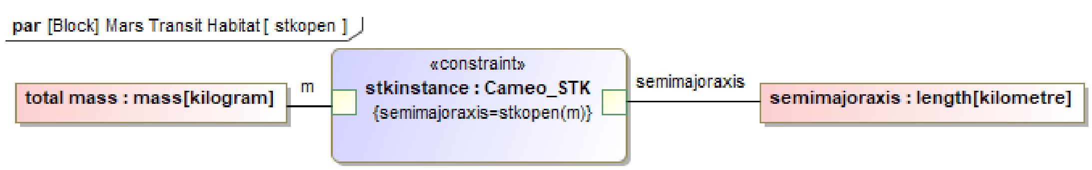

Figure 4: Parametric Diagram with MATLAB constraint block passing values from STK back into Cameo.

In the specific example set up by our developed models, the total mass for the Mars Transit Habitat calculated through for instance BDD in Figure 2 sets the satellite object mass in STK. Detailed astrodynamics information is beyond the scope of the DRA 5.0, but is certainly an essential component of Mars missions. While the MATLAB code in the Appendix defines sample orbit parameters [27], space engineers could alternatively model these parameters using MBSE techniques and then passed to STK's DE environment using Cameo's simulation capabilities.

B. Digital Engineering Results

AGI specifically designed STK for simulating space missions; its physics engine can model complex astrodynamics including Mars orbit insertion maneuvers and return to the user values for fuel load based on the object mass [23]. Figure 5 displays the STK visualization created from simulating Cameo BDDs with the MATLAB code in the Appendix; the visualizations shows a Mars Transit Habitat satellite and its 2D ground trace in a sample orbit.

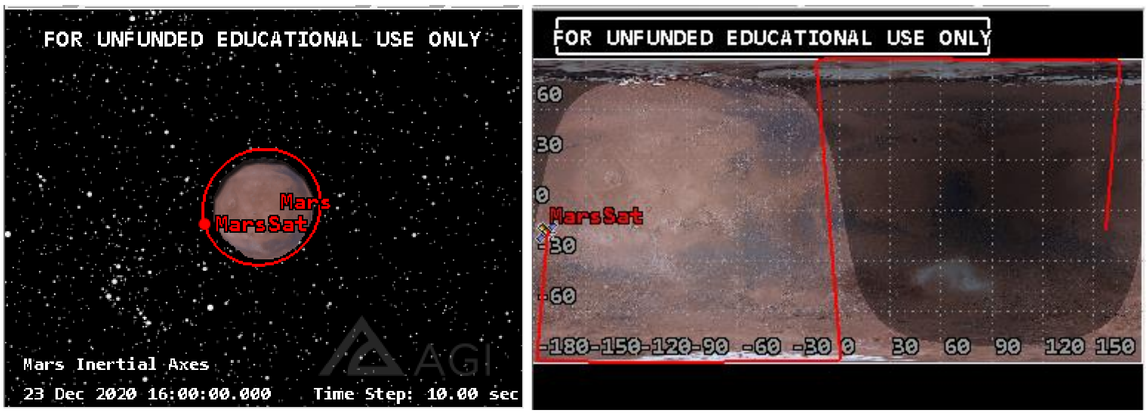

Figure 5: 3D and 2D STK graphics upon executing the BDD diagram in Figure 3 with MATLAB constraints.

Figure 6 demonstrates that interoperability was successful. The top image shows the results of the Figure 3 Cameo simulation using a MATLAB constraint block that populates the “semimajoraxis” value in the last few lines of the Appendix MATLAB code. The lower image shows the Mass property of the MarsSat satellite in STK, verifying that the value of 41,340 kg calculated by Cameo for the Mars Transit Habitat mass passed to STK.

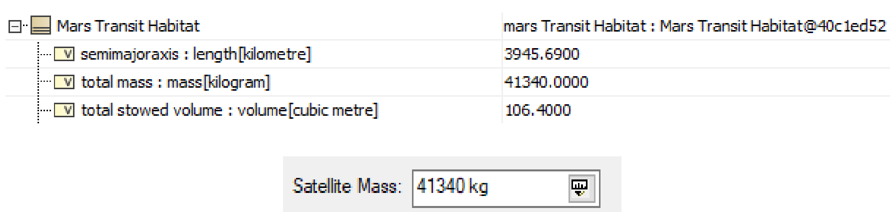

Figure 6: Screenshots from Cameo and STK demonstrating evidence that interoperability was successful.

Our MBSE/DE interoperability approach using a reusable MATLAB bridge improves upon manual movement of data from mutually exclusive MBSE and DE environments by decreasing the amount of time to simulate multiple system configurations. Furthermore, MBSE tools such as the ParaMagic Browser allow for fast editing of subsystem values, faster clicking individual objects in STK and changing parameters ore even editing the MATLAB code to change parameters [22]. Although using just MATLAB and STK provides many automation features, adding MBSE to that union provides context and organization. Creating this type of data automation pipeline remains a subject of further investigation. However, the code in the Appendix is a start to developing that type of automation process.

C. Assessing MBSE Benefits

The work by Henderson and Salado (2021), previously mentioned, lists the most stated MBSE benefit types found in literature [4]. To address that paper's point of studies lacking evaluation of these benefits in MBSE applications, we seek qualitative evidence of popular benefits in this crewed Mars mission modeling endeavor.

Table 2: Evaluation of MBSE benefit presence in our Mars mission plan development methodology.

MBSE Benefits [4]	Evidence
Higher level support for automation	The MATLAB interface combined with Cameo's simulation capabilities allows for DE process automation as demonstrated.
Reduce Ambiguity	SysML block names and formatting may be unclear to stakeholders unfamiliar with the language, but labels are prominent and understandable.
Early Verification and Validation	ParaMagic plugin can validate model correctness prior to execution. Organized block diagrams can aid manual model verification.
Increased efficiency	Model set-up time can be lengthy in the near-term, and we do not measure time savings in the long term to provide conclusive evidence for increased efficiency.
Better analysis capability	With MATLAB allowing interoperability between Cameo and STK, it is possible to analyze extensive systems that either software cannot capture individually.
Increased Productivity	The Cameo model building process is intuitive once learned and can help create STK scenarios faster and with reliable accuracy.
Improved System Understanding	Cameo's BDDs and Instance diagrams can provide clear descriptions of systems and their characteristics as demonstrated in Figures 2 and 3.
Reduce Errors	Cameo might not notice errors if the models are still validly structured. These errors can propagate throughout the model and into STK DE software.
Reduce Cost	While MBSE tools may come at a cost, the amount saved by MBSE's reduction of errors and completion time often leads to financial returns on this investment.
Improved Consistency	Because Cameo codifies relations between model blocks, BDDs maintain consistency in the system and subsystems.
Better communication/ information sharing	Diagrams such as Figure 3 provide system-level views that show stakeholders how elements interrelate; this can be more effective than traditional documentation.

Compared to MBSE and DE separately, there is not a commonality in literature of achievable benefits associated with the combination of the two fields together. The methodology proposed in this paper displays all the MBSE benefits as described in Table 2 and additionally touts the functionalities offered by DE, demonstrated in Figure 5. As such, our SysML, MATLAB, and STK methodology demonstrates improvements upon the process of developing crewed Mars landing missions using one of those languages individually; a novel contribution to this domain.

V. Conclusions and Future Work

We conclude this research with an evaluation of how our modeling endeavor confirmed or addressed the strengths and limitations of MBSE. This work has exhibited three SysML benefits: 1) SysML allowed descriptions of numerical quantities for standardized ISO units for mass; 2) SysML implements robust semantics by defining blocks in BDDs and value relations in parametric diagrams, the combination of which forms a system specification; 3) SysML models can symbolize a coherent body of knowledge about a system and interfaces with other tools, which we demonstrate with our methodology. We have addressed the three MBSE limitations previously described in this work: 1) Despite the lack of native support, our methodology achieves interoperability between MBSE and Digital Engineering using Cameo Systems Modeler, MATLAB, and Systems Toolkit; 2) Qualitative analysis provides evidence for MBSE benefits to respond to the dearth of this practice in literature; 3) We have raised awareness of the lack of cybersecurity considerations in MBSE for crewed Mars mission planning.

Our research discovered the following benefits to crewed Mars mission planning from integrating MBSE and DE:

1. Greater system simulation capabilities: STK scenario creation extends Cameo's simulations.
2. Enhanced model fidelity: physics-based values from STK can better inform SysML models.
3. Improved system understanding: STK graphics and data further build upon this MBSE benefit.

While this paper demonstrates an effective approach to linking Cameo and STK, research opportunities exist. For example, cybersecurity considerations of MBSE in crewed Mars Mission planning is a new area of exploration; future work can consider compatible ontologies and trust and reputation modeling to address limitations of past ongoing studies for developing space missions. Other research can focus on expanding the existing model, adding more details to subsystems, and increasing the number of components transferred to STK for analysis.

Now that we have demonstrated and discussed the benefits and future research required for successful integration of the fields of MBSE and DE, we return to our challenge: do space engineers need to learn MBSE to successfully adopt DE? While MBSE may not be necessarily a requirement for learning DE mission planning methods, it certainly boosts engineering capabilities. The field of MBSE/DE integration is growing: researchers are publishing more and more papers about interoperability between MBSE and DE. Industry is expanding on the use of this integration: AGI will soon update STK software with features for SysML integration. As MBSE/DE integration gains traction within the mission planning domain, crewed Mars missions shall become even more attainable.

Appendix

MATLAB code that when executed using the Cameo parametric evaluator creates a scenario in STK with a satellite orbiting Mars with a mass passed to it by Cameo. Propagates an orbit for the Martian satellite. To prevent Cameo from timing out while this code runs, change the simulation options by increasing external solver timeout.

```
function semimajoraxis=stkopen(m)
app = actxserver('STK12.application'); %Open STK 12
app.UserControl = 1;
app.Visible=1;
root = app.Personality2; %Grab a handle on the STK application root
scenario = root.Children.New('eScenario','MATLAB_Cameo_STK_Mars'); %Create new scenario
%Set scenario times
scenario.SetTimePeriod('23 Dec 2020 16:00:00.000','23 Dec 2021 16:00:00.000');
root.ExecuteCommand('Animate * Reset') %Reset animation times
mars = scenario.Children.New('ePlanet', 'Mars'); %Add Mars
%Add Mars ephemeris data
mars.CommonTasks.SetPositionSourceCentralBody('Mars', 'eEphemJPLDE');
% Control 2D graphics. 3D graphics inherit from 2D graphics automatically
mars2D = mars.Graphics;
mars2D.Color = 255;   % Set planet label to red
mars2D.Inherit = false; %Stop setting inheritance from scenario file
mars2D.OrbitVisible = false; %Delete marker for Martian orbit around Earth
mars2D.SubPlanetPointVisible = false; %Remove Mars point from  Earth
mars2D.SubPlanetLabelVisible = false; %Remove Mars label from Earth
%Switch central body to Mars on 2D graphics window
root.ExecuteCommand(['MapGraphics * SetCentralBody Mars']);
%Switch central body to Mars on 3D graphics window
root.ExecuteCommand(['VO * CentralBody Mars']);
%Add satellite to Mars
satellite = scenario.Children.NewOnCentralBody('eSatellite','MarsSat','Mars');
graphics=satellite.Graphics; %Control satellite graphics
graphics.Attributes.Color=253; %Set satellite label to red
satellite.MassProperties.Mass = m; %Set mass to received value
% IAgSatellite satellite: Satellite object
keplerian = satellite.Propagator.InitialState.Representation.ConvertTo('eOrbitStateClassical');
% Use the Classical Element interface
% Changes from Ecc/Inc to Perigee/Apogee Altitude
keplerian.SizeShapeType = 'eSizeShapeAltitude';
keplerian.LocationType = 'eLocationTrueAnomaly'; % Makes sure True Anomaly is being used
keplerian.Orientation.AscNodeType = 'eAscNodeLAN'; % Use LAN instead of RAAN for data entry
% Assign the perigee and apogee altitude values:
keplerian.SizeShape.PerigeeAltitude = 500;      % km
keplerian.SizeShape.ApogeeAltitude = 599;       % km
% Assign the other desired orbital parameters:
keplerian.Orientation.Inclination = 90;         % deg
keplerian.Orientation.ArgOfPerigee = 12;        % deg
keplerian.Orientation.AscNode.Value = 24;       % deg
keplerian.Location.Value = 180;                 % deg
% Apply the changes made to the satellite's state and propagate:
satellite.Propagator.InitialState.Representation.Assign(keplerian);
satellite.Propagator.Propagate;
%Return Semi Major Axis property
keplerian.SizeShapeType = 'eSizeShapeSemimajorAxis';
semimajoraxis=keplerian.SizeShape.SemiMajorAxis;
end
```

Acknowledgments & References

Special thanks to: Dr. Dirk A. Zwemer of Intercax, LLC for permission to use SysML models of "The Martian" infrastructure systems; Chairat Panpun of NoMagic, Inc. for aiding development in Cameo Systems Modeler; Dr. Eric Pearce, Dr. Alejandro Salado, and Dr. Mohammed Shafae for their reviews, insights, and suggestions.

[1] B. P. Douglass, "Chapter 1 - What Is Model-Based Systems Engineering?," in *Agile Systems Engineering*, B. P. Douglass, Ed. Boston: Morgan Kaufmann, 2016, pp. 1–39. doi: 10.1016/B978-0-12-802120-0.00001-1.
[2] S. Friedenthal and C. Oster, "Chapter 4 Applying SysML and a Model-Based Systems Engineering Approach to a Small Satellite Design," in *Advances in Systems Engineering*, American Institute of Aeronautics and Astronautics, Inc., 2016, pp. 127–218. doi: 10.2514/5.9781624104091.0127.0218.
[3] R. Wang and C. H. Dagli, "An Executable System Architecture Approach to Discrete Events System Modeling Using SysML in Conjunction with Colored Petri Net," in *2008 2nd Annual IEEE Systems Conference*, Apr. 2008, pp. 1–8. doi: 10.1109/SYSTEMS.2008.4518997.
[4] K. Henderson and A. Salado, "Value and benefits of model-based systems engineering (MBSE): Evidence from the literature," *Syst. Eng.*, vol. 24, no. 1, pp. 51–66, 2021, doi: https://doi.org/10.1002/sys.21566.
[5] L. Wang, M. Izygon, S. Okon, H. Wagner, and L. Garner, "Effort to Accelerate MBSE Adoption and Usage at JSC," in *AIAA SPACE 2016*, American Institute of Aeronautics and Astronautics, 2016. doi: 10.2514/6.2016-5542.
[6] J. W. Evans *et al.*, "Towards a Framework for Reliability and Safety Analysis of Complex Space Missions," in *19th AIAA Non-Deterministic Approaches Conference*, American Institute of Aeronautics and Astronautics, 2017. doi: 10.2514/6.2017-1099.
[7] Free, Jim, "Architecture Status." NASA Advisory Council Human Exploration and Operations Committee Meeting. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/march_2017_nac_charts_architecturejmf_rev_3.pdf
[8] J. Connolly, "Deep Space Transport (DST) and Mars Mission Architecture," p. 20.
[9] W. M. Cirillo, "Space Technology Mission Directorate Mars Architecture Technology Drivers Overview," p. 38.
[10] A. T. Klesh, J. W. Cutler, and E. M. Atkins, "Cyber-Physical Challenges for Space Systems," in *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, Apr. 2012, pp. 45–52. doi: 10.1109/ICCPS.2012.13.
[11] G. Falco, "The Vacuum of Space Cyber Security," in *2018 AIAA SPACE and Astronautics Forum and Exposition*, American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2018-5275.
[12] T. S. Ph.D, S. Visner, and S. C. K. Ph.D, "A Cyber Attack-Centric View of Commercial Space Vehicles and the Steps Needed to Mitigate," Nov. 2020, Accessed: Jun. 08, 2021. [Online]. Available: https://www.mitre.org/publications/technical-papers/a-cyber-attack-centric-view-of-commercial-space-vehicles
[13] G. Falco, "Cybersecurity Principles for Space Systems," *J. Aerosp. Inf. Syst.*, vol. 16, no. 2, pp. 61–70, 2019, doi: 10.2514/1.I010693.
[14] "Modeling to Mars: a NASA Model Based Systems Engineering Pathfinder Effort | AIAA SPACE Forum." https://arc.aiaa.org/doi/abs/10.2514/6.2017-5235 (accessed Feb. 24, 2021).
[15] S. Pavalkis, "MBSE in Real-Life Space Exploration Projects," *Modeling Community Blog*, Jul. 15, 2015. https://blog.nomagic.com/mbse-real-life-space-exploration-projects/ (accessed Nov. 04, 2020).
[16] S. David and F. Portree, "Humans To Mars: Fifty Years Of Mission Planning," *1950–2000 Nasa Monogr. Aerosp. Hist. Ser.*, vol. 21, 2001.
[17] B. B. Donahue, "Mars Transfer Vehicle and Lander Concepts for Human Exploration Missions in the 2031–2038 Time Frame," *AIP Conf. Proc.*, vol. 813, no. 1, pp. 1061–1070, Jan. 2006, doi: 10.1063/1.2169287.
[18] B. G. Drake, S. J. Hoffman, and D. W. Beaty, "Human exploration of Mars, Design Reference Architecture 5.0," in *2010 IEEE Aerospace Conference*, Mar. 2010, pp. 1–24. doi: 10.1109/AERO.2010.5446736.
[19] L. Anekwe, "Mars ship almost ready," *New Sci.*, vol. 243, no. 3250, p. 5, Oct. 2019, doi: 10.1016/S0262-4079(19)31837-8.
[20] "SpaceX: Mars & Beyond," *SpaceX*. http://www.spacex.com/human-spaceflight/mars/ (accessed Nov. 16, 2020).
[21] R. G. Merrill, P. Chai, C. A. Jones, D. R. Komar, and M. Qu, "An Integrated Hybrid Transportation Architecture for Human Mars Expeditions," in *AIAA SPACE 2015 Conference and Exposition*, American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2015-4442.
[22] "Modeling 'The Martian,'" *Intercax*, Oct. 16, 2015. http://intercax.com/2015/10/16/modeling-the-martian/ (accessed Nov. 21, 2020).
[23] "Digital Mission Engineering," *Agi*. https://www.agi.com/digital-mission-engineering (accessed Dec. 12, 2020).
[24] "MATLAB Interface." https://help.agi.com/stk/index.htm#matlab/matlab.htm (accessed Nov. 04, 2020).
[25] "AGI: Moxie." https://www.agi.com/products/moxie (accessed Dec. 12, 2020).
[26] D. Kaslow, B. Ayres, P. T. Cahill, L. Hart, A. G. Levi, and C. Croney, "Developing an MBSE CubeSat Reference Model – Interim Status #4," in *2018 AIAA SPACE and Astronautics Forum and Exposition*, American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2018-5328.
[27] "MATLAB Code Snippets." https://help.agi.com/stkdevkit/index.htm#stkObjects/ObjModMatlabCodeSamples.htm (accessed Jan. 02, 2021).