33rd Annual INCOSE
international symposium
hybrid event
Honolulu, HI, USA
July 15 - 20, 2023

# Orion SysML Model, Digital Twin, and Lessons Learned for Artemis I

Gregory J. Pierce – NASA's Orion Digital Twin Project Lead
NASA's Lyndon B. Johnson Space Center
2101 NASA Parkway
Houston, Texas 77058
281-792-5612
gregory.j.pierce@nasa.gov

Joshua D. Heeren – Orion Digital Twin Lead Model-Based Systems Engineer
JETS – Jacobs Engineering
NASA's Lyndon B. Johnson Space Center
2101 NASA Parkway
Houston, Texas 77058
281-244-9419
joshua.d.heeren@nasa.gov

Terry R. Hill – NASA's Digital Engineering Transformation Lead
NASA's Lyndon B. Johnson Space Center
2101 NASA Parkway
Houston, Texas 77058
281-483-8135
terry.hill-1@nasa.gov

**Abstract**. In 2015 it was recognized by NASA's Orion Chief Engineer that NASA's design insight into the Orion subsystems for Artemis I was not sufficient to provide standard engineering support to flight operations. To address these concerns, provide an opportunity to apply emerging model-based systems engineering and digital twin methodologies, and provide opportunities for employees across NASA to get hands-on training, an Orion Digital Twin pilot project was initiated in 2020 as part of the Agency's Digital Transformation initiative. With the increase in complexities of spacecraft, and decreased time to made decisions during missions in critical or emergency situations, digital modeling and integration of design can reduce the time to answer questions by days and the required human resources by an order of magnitude over conventional approaches and was identified to be a critical capability for NASA's future. This paper describes the genesis of the Orion Digital Twin pilot project, efforts undertaken, a reproducible methodology to take available system information from a mature program to create an executable SysML model that supports a link to the physical asset, and associated lessons learned and project deliverables.

# Introduction

The Orion Spacecraft is an essential component of NASA's Artemis missions to return to the Moon by transporting Astronauts to and from Lunar orbit (NASA 2023). The Orion Digital Twin provides an integrated schematics and simulation platform built to rapidly provide system information for the purpose of informing decisions within flight operational scenarios. This paper describes the genesis of this digital twin pilot project, efforts undertaken, a reproducible methodology to take available system information from a mature program to create an executable SysML model that supports a link to unit-specific data (forming a connection to the physical asset), and associated lessons learned and project deliverables.

Agency leadership saw the potential of digital twin technology applied to complex aerospace systems. Digital modeling and integration of design can reduce the time to answer questions by days and the required human resources by an order of magnitude over historical approach. Since there are currently few examples within NASA of spacecraft-scale digital twin applications it is challenging to demonstrate the value added by digital twins, resulting in early adopters facing a low level of technology maturity, and a lack of building-blocks to jumpstart development. This pilot project set out to mitigate those risks by demonstrating value by modeling the as-deployed design in comparison to traditional system engineering approach taken, and delivered examples of the methods, tools, and language needed to implement an effective digital twin of a complex aerospace system.

## *Background*

In 2015 discussions between NASA's Johnson Space Center Chief Engineer to Orion, the Program Office, and prime contractor regarding the design/build process for Orion Exploration Flight Test 1 (EFT-1), the need for vehicle schematics where end-to-end functionality could be traced was discussed, e.g. for: articulation of vehicle design, understanding functionality and interfaces, problem postulation and resolution. In addition to the limited Engineering schematic products, flight operations drawings (e.g., akin to Space Shuttle Systems Handbook/SSSH) were also not developed for EFT-1.

The issues, still outstanding, were raised again during the 2020 Artemis I System Acceptance Review (SAR) after which, the engineering directors at NASA's Langley Research Center and the Johnson Space Center approached the NASA Digital Transformation office and proposed a model-based system engineering (MBSE) SysML modeling approach which would likely provide the required technical design insight currently needed and a greatly enhanced access to information in addition to simulation / digital twin capabilities for real-time flight operations support. Project approval was granted in the fall of 2020 for an eighteen-month pilot to build a SysML-based digital twin of the Artemis I Orion electrical power system (EPS).

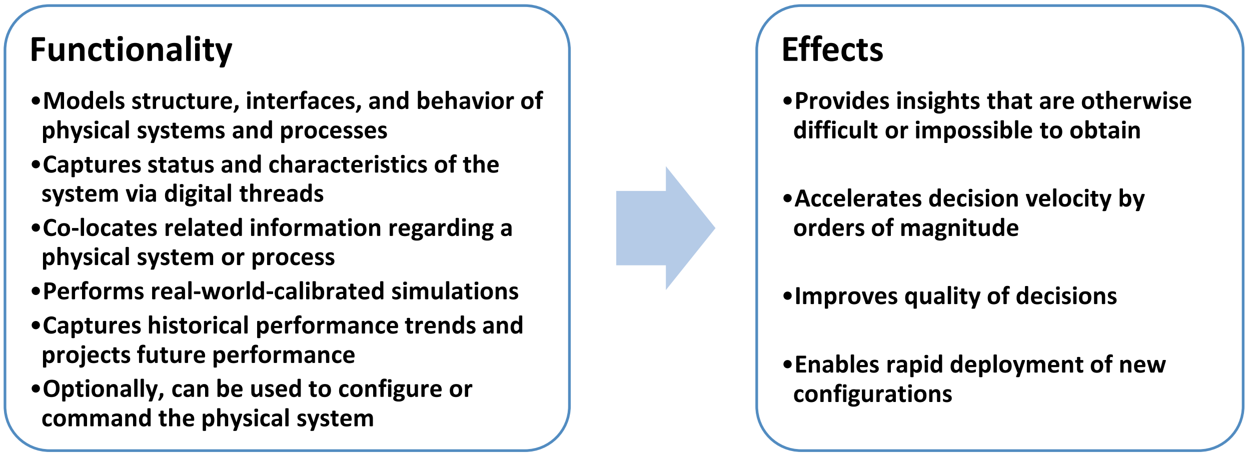
# The Digital Twin

One of the first challenges was centered around the definition of what a digital twin is or is not. The digital twin domain definition in 2020 was inconsistent across industry and government; therefore, to have a common place from all to start, the following definition the Digital Twin Consortium (DTC) was adopted for this effort.

> “A digital twin is a virtual representation of real-world entities and processes, synchronized at a specified frequency and fidelity.” (DTC 2022)

Notably, this requires the model to represent a physical system in the real-world. This is in contrast with other common uses of the term that focus more on high levels of design detail, 3D environments, metaverse implementations, or multi-domain integration. These were considered by the Orion Digital Twin team to be advancements in system modeling technologies but were distinct from digital twins. Another common point of variation among definitions were the required characteristics of the connection between the digital twin model and the physical asset. For example, the AIAA Digital Engineering Integration Committee (DEIC) discusses a connection that must be dynamically updated throughout its life cycle (AIAA DEIC 2020). While data updates done autonomously and frequently are highly desirable, such requirements would significantly and needlessly limit the application of digital twin approaches to inherently data-rich systems and characteristics of those systems. Many system development efforts would however benefit from system models that are calibrated with as-built, as-tested, and/or as-operated information. This includes the results of manufacturing inspections, 3D scans, initial acceptance testing, and annual checkouts. This also includes information from systems that lack the data infrastructure to support a dynamic update from an external system such as a digital twin.

Considering the above, the NASA Model-Based Engineering Digital Twin Team sought the functionality shown in Figure 1 to produce the listed effects.



**Functionality**

- **Models structure, interfaces, and behavior of physical systems and processes**
- **Captures status and characteristics of the system via digital threads**
- **Co-locates related information regarding a physical system or process**
- **Performs real-world-calibrated simulations**
- **Captures historical performance trends and projects future performance**
- **Optionally, can be used to configure or command the physical system**

**Effects**

- **Provides insights that are otherwise difficult or impossible to obtain**
- **Accelerates decision velocity by orders of magnitude**
- **Improves quality of decisions**
- **Enables rapid deployment of new configurations**

Figure 1. Digital Twin Functionality and Effects

Digital twin implementations are not specific to a single tool, language, or methodology. However, special emphasis was placed by the NASA Digital Engineering (DE) transformation team to utilize model-based systems engineering (MBSE), specifically SysML system architecture models, as the basis of digital twins, which could provide a consistent system engineering data thread throughout the lifecycle of a program/project. The ideal process is the creation of an architecture description model to support early life-cycle development, followed by the growth of that model to include executable simulation capabilities focused on project needs, and the eventual connection to physical assets (including prototypes, qualification units, and flight hardware) as they become available; see Figure 2.

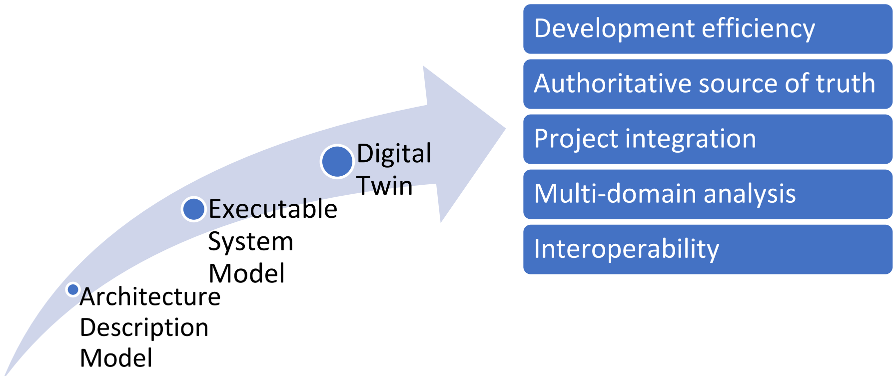


Figure 2. From System Model to Digital Twin

The intent is to encourage development efficiency by reducing the number of independent modeling efforts and to encourage an unambiguous authoritative source of truth (ASOT). This anchors the digital twin in the official system architecture description (to encourage project integration) by ensuring the modeling effort is a core element of the project development and encourages multi-domain analysis by leveraging the system model as the backbone for major system analyses and simulations. Further, desired interoperability is fostered by using tools, languages, and methods commonly employed across the organization of record.

## *Twininess*

Every digital twin is different. Factors include the intended application(s) for the digital twin, the unique characteristics of the physical system being modeled, and the technical capabilities being employed to make it happen. For many of these digital twin architecture dimensions, spectrums of possibilities exist, such as levels of autonomy and the extent to which the digital thread is utilized.

To help manage and discuss the vast array of possibilities, the NASA DE transformation team has employed the term "twininess" to refer to the extent, by both scope and degree, a system model is a digital twin. This includes the nature of the connection(s) to the physical system, the level of autonomy and frequency of synchronization, and the fidelity of the model to the physical system. Twininess also considers the use cases (or applications), system characteristics, and the underlying technical capabilities being employed; these later factors are summarized with examples in Figure 3, Digital Twin – Fingerprint.

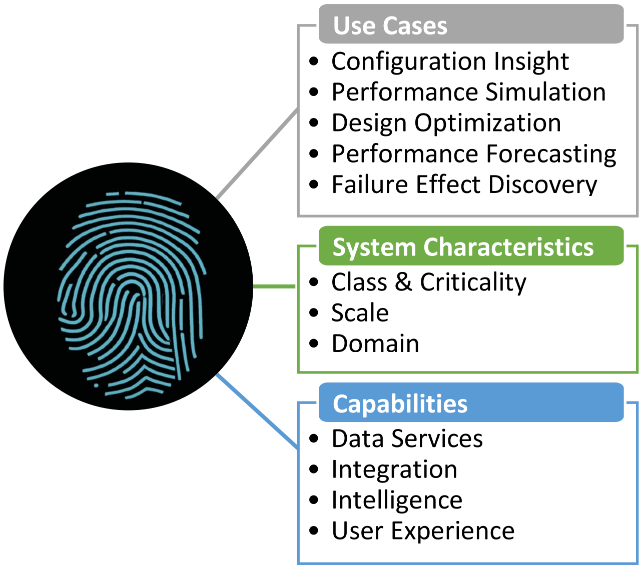


Figure 3. Digital Twin - Fingerprint

The project has outlined and categorized several typical use cases, many of which were derived from the list of examples listed by the

AIAA DEIC (2020) in their position paper on digital twins. The result is captured in Figure 4, which also highlights those primary cases targeted for the Orion Digital Twin work.

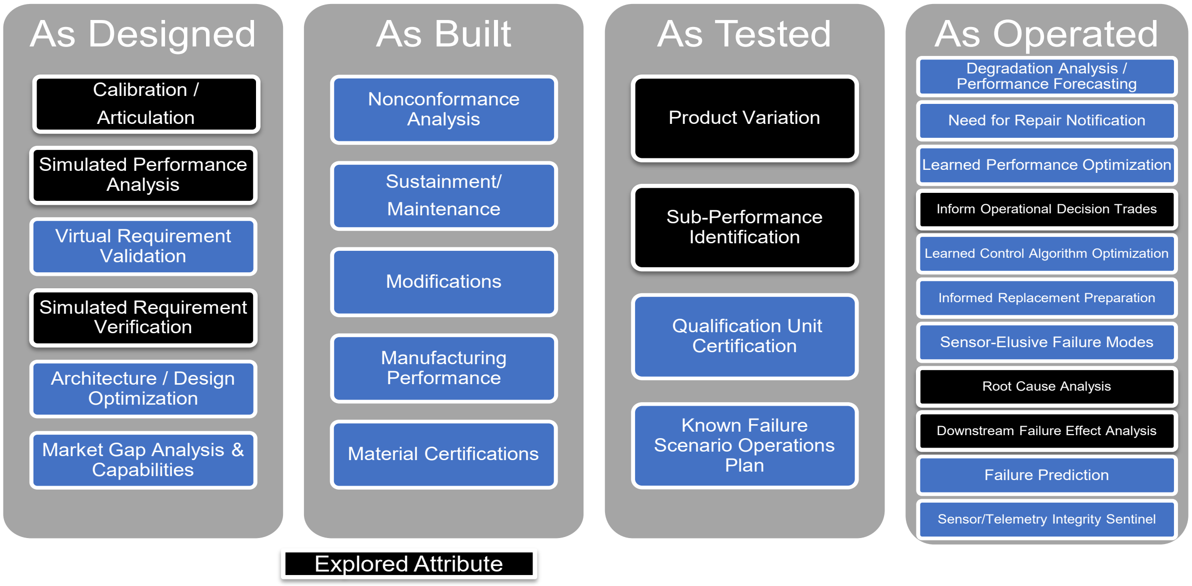


Figure 4. Use Cases that could be addressed by a digital twin, categorized by as designed, built, tested, or operated. Those shown in black were explored by the Orion Digital Twin Project.

Providing flight operations personnel and systems managers sufficient insight into the design and configuration of the vehicle was the primary need from the stakeholders. The team was able to enrich the capability with behavior models which simulate performance of Orion solar arrays, power distribution switches, and batteries.

The defining characteristics of the system being twinned have a significant effect on the development and capabilities that must be in place (or put in place). The Orion Digital Twin team (authors of this paper in addition to Anonymous at Org) analyzed the various dimensions of characteristics and settled upon those captured in Figure 5.

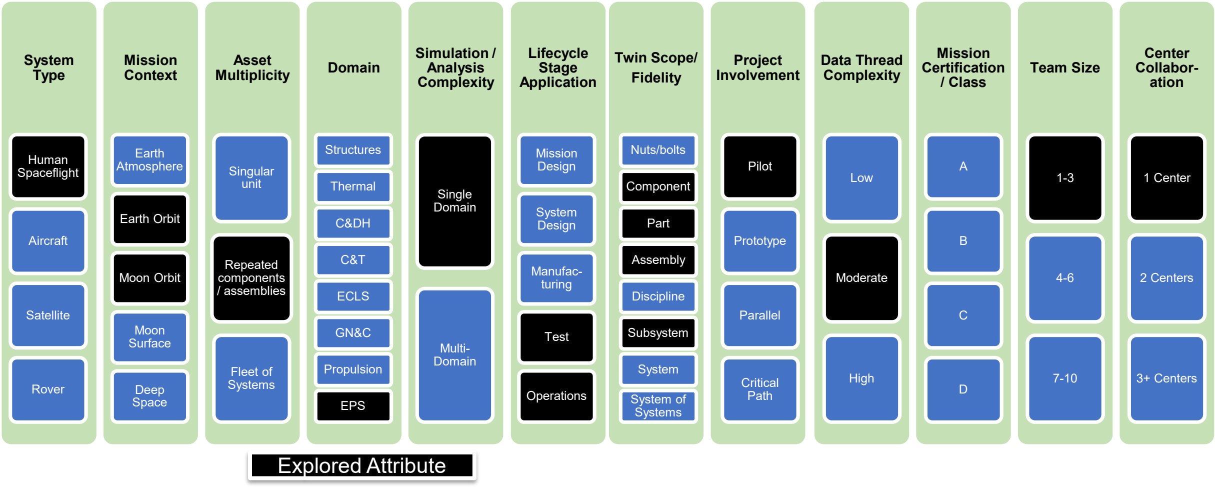


Figure 5. System Chacteristics to be considered when scoping the complexity and maturity of a digital twin. Those shown in black are the space explored by the Orion Digital Twin project.

The DTC (2022) released a listing of 62 technical capabilities, sorted into six categories (data services, integration, intelligence, user experience, management, and trustworthiness) in a product called the "Digital Twin Capabilities Periodic Table". As the required enabling capabilities vary greatly based on the factors above, those needed for the effort at hand can be readily identified. For Orion Digital Twin, the key identified DTC technical capabilities include 1. Data Acquisition & Ingestion, 3. Data Transformation, 4. Data Contextualization, 8. Data Aggregation, 22. API Services, 33. Simulation, 34. Mathematical Analytics, 39. Basic Visualization, and 42. Entity Relationship Visualization.

## Project Goals

The following were the overarching goals defined to benefit both the Orion Program and the NASA Engineering domain.

**Benefits To Orion:** The project sought to enable an enhanced human interface with data to enable increase data-driven decision velocity and accuracy. It should enable real-time mission support, decision making, and anomaly resolution with digital search via an intelligent application which has never been available via paper or PDF documents or drawings. Development of a robust SysML model and digital twin also provides an opportunity for the Orion Program, Engineering support, and flight operations communities to engage, and provide interface integration opportunities to Gateway, space suits, and international partner elements (Service Module, HTV-XG, etc.).

**Benefits To Systems Engineering:** Processes and capabilities defined and developed via this pilot would be made available to a broader Agency applicability and could be infused into a mainstream project when needed. The Orion Digital Twin would demonstrate the eventual evolution-target of a system architecture model being used in the operations phase of a system life cycle, similar to the evolution shown in Figure 2. The pilot would facilitate an integrated environment between modeled system, requirements, and baselined engineering models, schematics, and drawings. It would demonstrate model-based technical review capabilities which could be extended to project milestone reviews. It could also provide an environment where NASA personnel can get valuable, first-hand training experience with developing and interacting with system models.

## Approach

The Orion Digital Twin model would be initially developed as a pilot project with the potential for mission-support certification after the utility of the system is demonstrated for the Artemis-I configuration. As the Orion Program did not have a modern system model, the SysML model development of the Orion Digital Twin project started from a blank slate. Modeling a mature system also meant there was no opportunity to affect the design of the Orion to enhance the integration of a digital twin.

The initial challenge for the team was to develop a strategy to aggregate and transform available independent system artifacts (PDF document and drawings, Excel workbooks, PowerPoint and Word documents, etc.) in order to produce an integrated system architecture and simulation platform driven by the physical asset throughout the product lifecycle.

As an initial capability, the tool would capture interface and configuration data from all available design artifacts, reflect unit-specific operational data from ground testing and manufacture, and be able to ingest batched telemetry files as they are made available to the team.

While the massive collection of disjoint design artifacts created over the course of a decade was difficult to navigate, there were several rich sources of parsable data, including a complex spreadsheet of box-level (and some card-level) interfaces. The Orion Program's power budget spreadsheet was another vital source. These enabled rapid importation of both system components and interfaces and greatly expeditated system modeling. The available parsable information was not as detailed as desired, so significant levels of manual diagram generation was required.

The team did investigate a dynamic interface to the telemetry stream, but this was not feasible given the resources, the schedule placing Artemis 1 flight months away, and the project's status as a pilot.

The key functionality and deliverables from the engineering team included:

- Context-rich schematics providing the needed schematic information in addition to all levels detail (from highest level schematic down to hyperlinks to board-level diagrams) directly linked and easily navigable, in addition to being supported by a backend database which is queryable for design details which would not be available via traditional 2D schematic drawings.
- Web-accessibility – Users are able to access core system configuration information from a browser without the need for the installation of specialty software (simulations would still require a desktop client at this time). This interface also provides a platform for model-based reviews with the ability to add/reply comments on diagrams.
- Automated content (schematics), report generation, validation to authoritative sources, and data synchronization (updates component parameters used for runtime values in simulation).
- System-level integrated simulations: Interfaces, key parameters, mathematical relationships, requirements; component behavior; performance analysis, and verification of requirements.
- As-designed and unit-specific data which will provide the needed design insight for engineering support to flight operations.

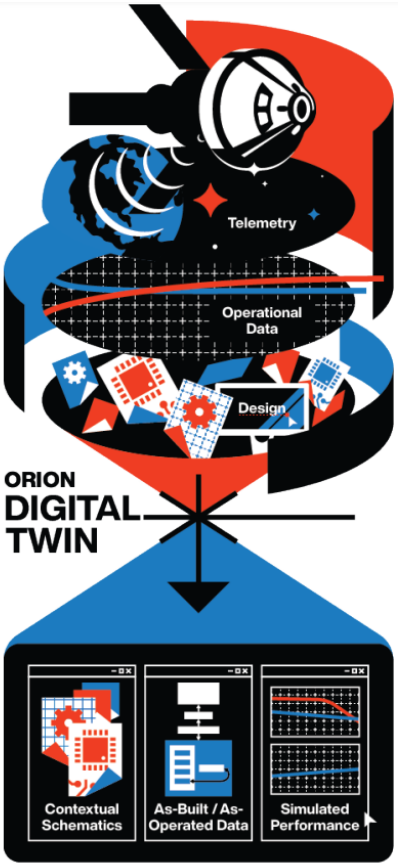


Figure 6. Orion Digital Twin

The team chose to utilize the system modeling tool MagicDraw® to create the foundation for a digital twin of Orion EPS which provided needed insight. While development began with the EPS and with the goal of a validated Orion EPS Digital Twin to be available during the Artemis 1 mission, subsequent development would continue with additional Orion subsystems, or it could also proceed with the implementation of additional digital twin capabilities and simulation use cases.

As a pilot implementation, formal certification Orion EPS Digital Twin for flight mission support was not required. Verification was performed by the team to ensure the technical content of the model accurately reflected the baselined design and the hardware configuration. For parsable data-sources, autonomous software scripts were developed to highlight differences between model and source artifacts. These scripts were verified to function correctly via testing and code-review. For non-parsable data-sources, the team performed several rounds of both peer and subject-matter expert reviews of model content.

## Artemis 1 Mission Validation

The Orion Digital Twin team aimed to both provide potential value to the Orion Program for the Artemis 1 mission and to receive valuable mid-mission feedback on the use of the tool by the Orion EPS System Management (SM) team. A series of evaluations was performed with the SM team, each of which consisted of a hypothetical anomaly situation and utilized the Cameo Collaborator web application. The graphical comment feature in Collaborator was used to flag the anomaly, which were then accessed by the evaluator on the comments panel. This demonstrated an efficient means of communication between flight controllers and system managers, as this built-in feature directly connects the user to flagged model elements for quick resolution. The evaluators then used the dynamic navigation capabilities of the tool to explore the system for possible proximate causes for the anomaly as well as quickly accessing detailed design artifacts that were linked to the model. Each session ended with a short set of interview questions. The evaluators found the tool powerful but approachable and especially liked the ability to dynamically navigate the system. Overall, the Orion Digital Twin did meet its objective of improving the ability to rapidly gain insight into system design and configuration.

## Pilot Project Accomplishments

At the completion of the eighteen-month pilot project, the following metrics and deliverables were provided back to the Agency. The bulk of the core model was built by a full-time modeler over a five-month period; this does not include an initial development and prototyping period, and it was followed by extensive investigations into digital twin integrations and simulation applications.

**Model metrics:** The SysML model of the EPS contains over 9,000 components represented (~600 elements of definition reused, 130+ hyperlinks to schematics database), ~4,500+ interfaces defined between components, 500+ loads integrated into EPS architecture (channelization to specific switch within power distribution units (PDUs), individual load parameters (‘spec power’, ‘actual power’, etc.).

**Core Capability:** Interfaces from PDUs and power conditioning and distribution units (PCDUs) to other subsystems. All EPS components, PDU/PCDUs, and avionics boxes are represented as loads on the EPS are detailed within the SysML model.

**NASA peer reviewed on August 2, 2021:** Participants included MBSE Infusion and Modernization Initiative (MIAMI) leaders from across the Agency, members of the NASA MBSE Leadership Team, and experienced modelers from MBSE vanguard projects at Jet Propulsion Laboratory, NASA’s Goddard Space Flight Center, Glenn and Langley Research Centers. Feedback was very positive overall and gained validation the approach avoided some common pitfalls. Comments emphasized the importance of reliable data interfaces and model validation tools.

**Deliverables:** Orion Digital Twin Model – Validated and functional digital twin of the EPS, including schematics from system to intra-card details, simulation capability, web access, as-built and as-tested data, and telemetry interfaces. This has been made available to Orion Program personnel for consumption and use only.

The following deliverables were made available to the NASA MBSE Community via NASA’s Office of Chief Engineer (OCE) managed model repositories as a template and example for general use. Some of this content may be available for public release; those interested should contact the authors.

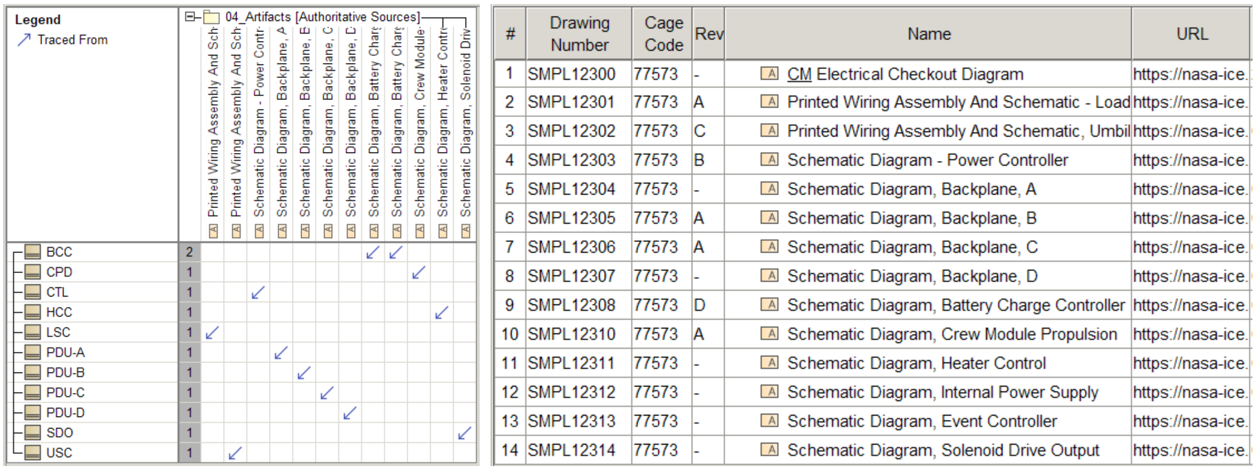
- Sanitized Spacecraft Digital Twin Model – Generic spacecraft system digital twin model based on the Orion Digital Twin content but striped of any controlled information.
- Library of Model Elements – Sanitized model elements, such as block and part definitions of components such as batteries, power switches, and solar arrays.
- Library of Automation Scripts and Functions – Collection of software code developed for this effort. These are primarily functions, captured as in-model opaque behaviors, for the utilization of the MagicDraw API (application programming interface). This includes the import of data from external sources and the creation and manipulation of model elements.
- Digital Twin How-To Guide – Document capturing the process a project modeling team can follow to develop of design and configuration digital twin of their system.
- Documented Orion Digital Twin Lessons Learned, Tips, and Tricks – Compilation of lessons learned, tips, and tricks identified during this and might be of use to any modeling team developing a digital twin or an MBSE system model.

## Lessons Learned and Best Practices from Orion Digital Twin Modeling

### *Synchronization with Design Artifacts*

As the detailed technical content of each source artifact was interpreted and recreated in the model, the project recognized the configuration management challenge being created. This led to the priority placed on parsable data sources and the generation of MagicDraw API scripts, discussed below, to automatically ingest data from sources, and generate appropriate model elements. Additional scripts were developed to allow the comparison of model elements from an updated data source. Programming the scripts to directly make model updates was explored and demonstrated as well, but this was not recommended on a previously laid-out diagram as automatic changes can lead to significant auto-generated-diagram clean-up and thus the team preferred making manual changes as needed.

For non-parsable documents and sources, the team relied on an in-model index of source artifacts, including key configuration management information, and references to the primary storage location. As they were model elements themselves, they could be linked to the blocks they defined, as shown in Figure 7.



| | | Printed Wiring Assembly And Sch | Printed Wiring Assembly And Sch | Schematic Diagram - Power Contr | Schematic Diagram, Backplane, A | Schematic Diagram, Backplane, B | Schematic Diagram, Backplane, C | Schematic Diagram, Backplane, D | Schematic Diagram, Battery Char | Schematic Diagram, Battery Char | Schematic Diagram, Crew Module | Schematic Diagram, Heater Contr | Schematic Diagram, Solenoid Driv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BCC | 2 | | | | | | | | ↙ | ↙ | | | |
| CPD | 1 | | | | | | | | | | ↙ | | |
| CTL | 1 | | | ↙ | | | | | | | | | |
| HCC | 1 | | | | | | | | | | | ↙ | |
| LSC | 1 | ↙ | | | | | | | | | | | |
| PDU-A | 1 | | | | ↙ | | | | | | | | |
| PDU-B | 1 | | | | | ↙ | | | | | | | |
| PDU-C | 1 | | | | | | ↙ | | | | | | |
| PDU-D | 1 | | | | | | | ↙ | | | | | |
| SDO | 1 | | | | | | | | | | | | ↙ |
| USC | 1 | | ↙ | | | | | | | | | | |

| # | Drawing Number | Cage Code | Rev | Name | URL |
|---|---|---|---|---|---|
| 1 | SMPL12300 | 77573 | - | CM Electrical Checkout Diagram | https://nasa-ice. |
| 2 | SMPL12301 | 77573 | A | Printed Wiring Assembly And Schematic - Load | https://nasa-ice. |
| 3 | SMPL12302 | 77573 | C | Printed Wiring Assembly And Schematic, Umbil | https://nasa-ice. |
| 4 | SMPL12303 | 77573 | B | Schematic Diagram - Power Controller | https://nasa-ice. |
| 5 | SMPL12304 | 77573 | - | Schematic Diagram, Backplane, A | https://nasa-ice. |
| 6 | SMPL12305 | 77573 | A | Schematic Diagram, Backplane, B | https://nasa-ice. |
| 7 | SMPL12306 | 77573 | A | Schematic Diagram, Backplane, C | https://nasa-ice. |
| 8 | SMPL12307 | 77573 | - | Schematic Diagram, Backplane, D | https://nasa-ice. |
| 9 | SMPL12308 | 77573 | D | Schematic Diagram, Battery Charge Controller | https://nasa-ice. |
| 10 | SMPL12310 | 77573 | A | Schematic Diagram, Crew Module Propulsion | https://nasa-ice. |
| 11 | SMPL12311 | 77573 | - | Schematic Diagram, Heater Control | https://nasa-ice. |
| 12 | SMPL12312 | 77573 | - | Schematic Diagram, Internal Power Supply | https://nasa-ice. |
| 13 | SMPL12313 | 77573 | - | Schematic Diagram, Event Controller | https://nasa-ice. |
| 14 | SMPL12314 | 77573 | - | Schematic Diagram, Solenoid Drive Output | https://nasa-ice. |

Figure 7. Source Traceability Matrix (sample content) linked to model blocks

## Artifact Navigation

The team's approach demonstrated the value of co-located, dynamically navigable information using out-of-the-box functionality in Cameo Collaborator®. In an operations troubleshooting scenario, system schematics are used to characterize and resolve anomalies by identifying functional dependencies throughout a system. The status quo of document-centric methods involves the traversal of hundreds of disjoint schematic PDFs which rely on the user's ability to follow interface "breadcrumbs" across the various documents, spreadsheets, and drawings. For example, to discover which systems are involved with supplying power to an anomalous load, an engineer might (a) look up that load on the electrical equipment list, (b) consult the program database of electrical connections, (c) find the schematic for the switch card the load is most directly plugged into, (d) identify which connections to search for in a backplane configuration drawing, (e) identify the correct paths to follow in the set of harnessing drawings, and (f) repeat until the appropriate switch is determined. This process is *time consuming and error prone*. MBSE has long benefited the design process by having interconnected information at the foundation of the practice. By leveraging this attribute of SysML for the benefit of flight operations personnel, the manual effort required to utilize system schematics as a whole is reduced to simple and intuitive clicks. The interconnectivity of a complex system is baked into a traversable web of system schematics which allows the user to follow an electron from tip to tail.

## Human Error Reduction

During the course of developing the Orion Digital Twin, errors were discovered in the source design baselined material (thus demonstrating the value of systems modeling inherent to SysML with a real-world example) including but not limited to:

- Solar Panel parameters within design document had mismatched columns.
- Discrepancies between pages within electrical checkout diagrams between physical wire pin-in pin-out mates.
- Appearance of copy-and-paste signal designators lacking necessary changes within new context.
- Signals labeled as ground on both power and return interfaces.
- Inconsistent updates to design changes within a single document.

## *Parametric Equation Solving*

**Deconflicting human-readable diagrams and MagicDraw Equation Solving:** One of the tenets of modeling adopted by the team was to construct the model such that it reflects the system of interest. While this seems obvious, it is often tempting to use SysML constructs in creative ways in order to achieve some desired outcome, such as getting a simulation to produce valid numbers by misrepresenting the system architecture. This is using the MBSE tool to program a simulation, whereas the ideal implementation of MBSE is to describe how that system is constructed and behaves and then execute that model to obtain valid results. One example encountered by the team was the drawing of power by an electrical load from a power bus. A human reader, especially a 'non-modeler', would expect to see 'power' being provided to the load from the bus and all arrows pointed in directions consistent with that. Due to the fact that the simulation engine in MagicDraw, Cameo Simulation Toolkit (CST), does not consider item flows, the team relied on flow properties of proxy ports to depict power interfaces and the flow of electricity between them.

In order to propagate changing values of component variables throughout the simulated system, CST implements flow properties with "in" vs "out" characteristics. An "out" flow property within a proxy port of a block can provide changes to its variable values to other blocks. This directional flow of value changing, which is necessary to simulate the system, does not always align with a human understanding of the 'flow' of interfaces. But in implementing the executable model it became necessary to have the electrical bus provide voltage value changes to the load which in turn replied to the electrical bus with the associated current to meet the load's power specification.

This scenario required voltage "out" and current "in" flow properties at the edge of the electrical bus block. To better preserve the SysML model's capability to communicate system design to human readers, two separate conceptual layers of flow properties were needed to both depict the direction of component interfaces for human understanding and to provide viable executable SysML running behind the scenes. This is a non-ideal situation and further specifications of executable SysML should take this into account to eliminate this disparity between model execution and human understanding.

**Functional Mock-up Unit (FMU) Integration:** In combining encapsulated executable elements, the dependencies between component value properties through their interfaces may not always be apparent from the start. The system simulation as a whole must be integrated such that during each step of time progression the resultant system of equations arrives at a solution for all variables involved. Particular attention must be paid to the order in which component variable calculations and associated values propagate throughout the system. CST updates value changes one-by-one as the components receive input values and produce output values. Variable value updating can be trapped within a loop which causes the simulation to not advance to the next step until the last variable no longer receives an updated value. The propagation of value updating is analogous to a traditional feedback loop control system.

Much like control theory dictates, looping situations within a co-simulation can either reach convergence, diverge to arbitrarily large numbers, or pivot around a pseudo-stability point. These situations can be remedied by reorganizing the encapsulation of executable model elements to be fully within a single FMU which is able to symbolically solve the system of equations, providing bounds to acceptable resting variable values within a single time-step, delays to inputs within a loop, or aiding the looping iteration by introducing virtual control mechanisms such as a damping feature in between component variables related across component boundaries. These control mechanisms do not alter the steady-state values in relation to time-stamps but help the system level simulation reach convergence within a single time-step. This allows CST to effectively solve systems of equations without a 'solver' by utilizing input-output simulation constructs.

**MATLAB Integration:** MagicDraw includes the capability to integrate MATLAB or Simulink files into the executable behavior of blocks. When implemented in a large complex system such as the Orion EPS, it slowed the simulation considerably when the CST master algorithm governing the execution of system components had to sequentially call the MATLAB script multiple times within a single time-step. Though the use of MATLAB files in the system simulation was largely abandoned for the Orion Digital Twin, some efficiencies could be gained by feeding a matrix of component values into a singular MATLAB file which would only need to be run once per time step for multiple components. This approach gains efficiencies in simulation speed but loses efficiencies in component re-use and manual creation of extra interfaces from the MATLAB to the system components in SysML.

## *Logical Inheritance*

The definition of a 'logical model' is a contentious subject among systems engineers. This paper does not attempt to prescribe a definition on the concept but does use the term to describe a convenient layer of system architecture as it is applied in this methodology. In relation to the normal flow of systems engineering activities throughout a typical project lifecycle, this project worked backwards from available physical system documentation to appropriately abstract functional and logical layers in executable SysML. Functional SysML blocks were abstracted such that they were able to be executed as a black box and then integrated into logical groupings of functions and then 'allocated' to the appropriate physical block through inheritance. This allows the physical blocks to receive updated values across physical interfaces, bind those physical interface properties to the logically grouped functional interfaces and then simulate component behavior as directed by the logical blocks to then supply physical interface outputs to their destination in the physical architecture. This logical layer provided a means to conveniently tailor the level of fidelity in simulation to efficiently provide the desired insight into integrated component behavior and performance while considering the available input values that could be utilized from the physical asset telemetry. Additionally, this maximized the utility of SysML element reuse by characterizing repeated functionality that can be implemented in various configurations throughout the physical architecture.

As indicated by the utility of reverse engineering functional and logical layers to construct a digital twin, maintaining a SysML architecture from concept through operations will not only streamline development processes but also provide a naturally evolving framework for reaching digital twin end goals.

## *In-Model Scripting with Opaque Behaviors*

The automation software scripts used to create and maintain the SysML structural elements from parsable authoritative sources were developed, organized, and implemented within the model itself. Generic tables provided a convenient presentation of the reusable scripts captured in SysML opaque behaviors which could then be dragged-and-dropped onto activity diagrams to develop more complex algorithms. Cameo Simulation Toolkit can execute those activity diagrams running the scripts to perform the model manipulation needed to first generate and then maintain the synching of the digital twin model to changing source data.

## Forward Work

As applications and implementations of digital twins are very diverse, so is the myriad number of next-step opportunities. In order to manage and bound the issue, and to ensure progress is made towards Agency goals, the team is leveraging the twininess and fingerprint concepts discussed previously. Exploring different system types (e.g. an aircraft) or different mission contexts (e.g. the surface of the Moon) were considered, but it was determined these changes would not teach new lessons beyond proving the extensibility of the techniques and building a digital twin via different system type would require spending time and resources to reproduce capability demonstrated by the Orion Digital Twin first in order to then expand the experience with other twin system characteristics as defined in Figure 5. Instead, focus will be placed on leveraging content already in place as a platform for incorporating new levels of complexity and ultimately improved levels of twininess.

**Mission Certification and Artemis 2 Update for EPS:** Pending the results of the Artemis 1 mission evaluation activities, the Orion Digital Twin project will propose to the Orion Program that the model be updated to reflect the Artemis 2 configuration and a process for certifying the tool for mission support.

**Co-Simulation of Subsystems:** Extend the application of approach to additional subsystems and eventually the full spacecraft. The Orion subsystem of greatest interest to the team is Guidance, Navigation, and Control due to the possibilities for co-simulation of solar array performance with vehicle attitude and integration with mission environment simulations.

**Mission Systems Integration:** Integrate the Orion Digital Twin model into the main telemetry/data system for the Orion Program. This would realize the live/dynamic synchronization of the model to the physical asset. Telemetry would be used both to capture the state of the system (actual power loads, switch states, etc.) as well as to calibrate the model with the latest system performance characteristics using AI/ML (artificial intelligence and machine learning) techniques.

**Pause-and-Play Capability:** Develop and demonstrate the ability to execute a hand-off between telemetry-based status and predictive simulation.

**Full Requirements Implementation:** Continue the integration of Orion Program requirements and specifications into the digital twin behavior models to provide full traceability between model constraints and the Program's ASOT as well as inform new and critical simulation use cases.

**Operational Data Systems Integration:** Integrate the Orion Digital Twin model into the ASOT data sources of operational data, including discrepancies, acceptance and lot testing, hardware scans to populate structural parameters, etc.

**Volumetric Model Integration:** Integrate the Orion Digital Twin model, which is primarily interfaces and behavior elements, with 3D CAD (computer-aided design) models and the associated ASOT repositories. This would connect two key aspects of the integrated system model as well be a key enabler to eventual metaverse integration.

**Learned Fault Recognition and Prediction:** Identify markers that could lead to co-morbid faults in the Orion EPS digital twin using AI/ML analysis methods.

## Summary

The initial development of the Orion EPS Digital Twin is complete and was ready to assist in mission support activities for the Artemis 1 mission. The Orion Digital Twin team has successfully demonstrated that a detailed digital twin system model can be created and maintained late in the development lifecycle, in addition to provide design insight in support of flight operations. The team has successfully shown that aggregating as-designed, as-built, as-tested, and as-operated data can provide insights not possible using traditional document-based approaches, and the time needed to answer critical questions is reduced by orders of magnitude. The Orion Digital Twin model is also an excellent validation of MBSE's promised value of reuse, having represented over 9000 components with only 600 definition elements in the model. The building-blocks the team set out to create and help jump-start new modeling efforts are completed and are already seeing use by major NASA programs, such as Gateway. The Orion Digital Twin is also already having the desired effect of inspiring and emboldening projects within NASA to undertake system architecture modeling, MBSE-based simulation, and digital twin development as it is now a proven, efficient, and effective approach to solving NASA's technical challenges.

## References

AIAA DEIC 2020, "Digital Twin: Definition & Value – An AIAA and AIA Position Paper", American Institute of Aeronautics and Astronautics (AIAA), December 2022.

Digital Twin Consortium (DTC) 2022, "Capabilities Periodic Table – Digital Twin Consortium", DTC, viewed 01 November 2022, <https://www.digitaltwinconsortium.org/initiatives/capabilities-periodic-table/>.

DTC 2022, "Definition of a Digital Twin – Digital Twin Consortium", DTC, viewed 01 November 2022, < https://www.digitaltwinconsortium.org/initiatives/the-definition-of-a-digital-twin/>.

National Aeronautics and Space Administration (NASA) 2023, "NASA Artemis", NASA, viewed 01 April 2023, < https://www.nasa.gov/specials/artemis/>.

# Biography

**Greg Pierce** is the Model-Based Systems Engineering Lead for the Engineering Directorate at NASA Johnson Space Center, working to improve the institutional compatibility and grow the practice at JSC and the Agency. He also contributes to the leadership of the systems engineering discipline at JSC in several capacities, such as chairing the JSC Systems Engineering Working Group. Before joining NASA directly in 2020, Pierce supported NASA JSC for over 15 years as a support contractor with Jacobs, where he led their MBSE Initiative. Pierce is a career systems engineer with extensive experience in the development of flight system such as space suits and jetpacks all while seeking opportunities apply or innovate MBSE approaches to efficient and effective execution of projects. Pierce, a long-time INCOSE CSEP and member, has a bachelor's in aerospace engineering from the University of Michigan in Ann Arbor.

**Joshua D. Heeren** is the Lead Model-Based Systems Engineer on the Orion Digital Twin project, as an employee of Jacobs on the JETS (JSC Engineering, Technology, and Science) Contract, drawing from extensive experience in implementing model-based approaches to systems engineering challenges across NASA and the DoD on spacecraft, aircraft, and business enterprise architectures. He has a bachelor's in aerospace engineering and a bachelor's in psychology from the Georgia Institute of Technology.

**Terry R. Hill** serves as the Lead of the NASA Digital Engineering Transformation and the Model-Based Systems Engineering Leadership Teams, which are responsible for providing recommendations for a strategic and implementation approach to deliver digital engineering and MBSE methodology and interoperable toolchains to usher the agency into the modern world of data-centric design and systems engineering. Hill is the assistant division chief of Johnson Space Center's Engineering Directorate's Project Management and Systems Engineering Division and previously held positions as the next-gen Constellation Space Suit Engineering Project Manager; International Space Station Extravehicular Mobility Unit Deputy Subsystem Manager; and Deputy Program Manager of the Crew Health and Safety Program. He has a bachelor's in aerospace engineering and a master's in aerospace guidance, navigation and control theory from the University of Texas at Austin.