DEEP-SPACE HABITAT AUTONOMY SUPPORTED BY HIERARCHICAL DIGITAL TWIN ARCHITECTURE

By

CORY ALLAN GEORGE

THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

MECHANICAL AND AEROSPACE ENGINEERING

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Stephen K. Robinson

Mario Berges

Zhaodan Kong

Committee in Charge

2023

Table of Contents

Acronyms

ACAWS	Advanced Caution and Warning System
ARS	Atmosphere Revitalization System
ASL	Autonomous Systems Laboratory
BIM	Building Information Model
CDRS	Carbon Dioxide Removal System
DAQ	Data Acquisition
DCR	Downward Counterfactual Reasoning
DRM	Design Reference Mission
ECLSS	Environmental Control and Life Support System
EDT	Element Digital Twin
EPS	Electrical Power System
ESM	Element Systems Manager
FSRM	Fault-Symptom Relationship Model
GUI	Graphical User Interface
HOME	Habitats Optimized for Missions of Exploration
HVAC	Heating, Ventilation, and Air Conditioning
IOH	Integrated Orbital Habitat
IoT	Internet of Things
ISHM	Integrated System Health Management
ISS	International Space Station
LEO	Low Earth Orbit
MAST	Modular Autonomous Systems Technology
MBSE	Model-Based Systems Engineering
MC	Mission Controller
MCED	Model Conformity Enforcement Driver
M&R	Maintenance and Repair
MTM	Maintenance Task Management
MOH	Mars Orbital Habitat
MTV	Mars Transit Vehicle
NPAS	NASA Platform for Autonomous Systems
OODA	Observe, Orient, Decide, Act
PDM	Power Distribution Module
PPE	Power and Propulsion Element
RCA	Root Cause Analysis
RT	Research Thrust
RUL	Remaining Useful Life
SDT	System Digital Twin
SPM	System/Process Manager
STRI	Space Technology Research Institute
TCS	Thermal Control System
THCS	Temperature and Humidity Control System

UPA	Urine Processing Assembly
VDT	Vehicle Digital Twin
VSM	Vehicle Systems Manager
WIE	What-If Engine
WRS	Water Recovery System

List of Figures

List of Tables

Acknowledgements

This work represents the culmination of hard work and schooling that spans a decade. There were so many people that have helped me along this path, I could not possibly list them all. From all the professors, mentors, and colleagues at Los Angeles City College and UC Davis to all the folks at NASA and in the HOME community that taught me so many valuable lessons. To all my family and dear friends that cheered me on and convinced me not to quit… Thank you all from the bottom of my heart.

Abstract

The next generation of human-rated space vehicles will need to keep humans alive further from Earth than ever before. This prospect introduces many unique challenges including issues with logistics, communications, and safety. The nature of deep-space habitation will necessitate a level of vehicle autonomy not yet realized for human-rated spacecraft. This work proposes that Digital Twin technology can be used as a new tool to enable heightened autonomy and situational awareness that will ensure the safety of the vehicle and its crew. Use-cases specific to deep-space habitation are explored in detail. A Digital Twin Framework capable of supporting deep-space habitat autonomy is proposed and applicable engineering requirements are established. A demonstration implementation is proposed as a proof of concept to show the basic functioning and utility of the proposed Digital Twin. This study found that a Digital Twin is a necessary and enabling piece of technology for future human-rated spacecraft venturing into deep-space. Necessary future research targets are identified including reduced-order system-of-systems modeling, cross-scale coupling of models, integrated model verification and validation, and semantic interoperability that allows clear communication of any data, information, or query between the Digital Twin and its users.

Introduction

Space is immensely hazardous to Human life. Space is remote, pressures are near total vacuum, there is intense radiation, and relative velocities between objects can turn a ping pong ball into a bullet. Arguably the most insidious hazard of all is the lack of infrastructure. If a safety-critical component of a vehicle fails, it cannot be towed to a mechanic. You must either fix it with what you brought with you or try to make it home before some critical-to-life condition is reached. These hazards have driven a largely incremental approach to Human spaceflight. Each step has been taken with the utmost consideration for safety. It is essential to assure that the brave men and women who enter this environment have the best chance of making it home safely.

With the exception of the Apollo program, Human spaceflight has been constrained to Low Earth Orbit (LEO). LEO is close by, meaning that in an emergency, crew can get home relatively quickly. The International Space Station (ISS) orbits at an average altitude of 400 km and, as of this writing, has been continuously occupied for nearly 22 years. Currently, sensor data from systems onboard the ISS are transmitted to Mission Controllers (MC)s on the ground. There are onboard threshold-related alarms associated with critical systems, but most of the monitoring of the state of health of systems is done by human experts on Earth. Those MCs are specialized to understand the telemetry coming in from those systems. It is their job to interpret the state of health of their assigned systems, predict their Remaining Useful Life (RUL), and recommend mitigations or repairs to the crew onboard the vehicle. This system has worked well for LEO where there is near real-time communications including real-time telemetry of data from thousands of sensors, strong infrastructure for regular resupply, and a quick trip home in an emergency.

Beyond LEO, the game changes. Other massive bodies begin to enter the gravitational equation and the time and energy cost of resupply or emergency return is entirely dependent on the laws of orbital mechanics and rocket propulsion. Perhaps the most famous example of this situation was the Apollo 13 mission. On the way to the Moon, an unforeseen electrical failure caused an oxygen tank to explode. The trajectory of the spacecraft meant that it could not simply turn around and head home. It had to complete its trip around the Moon before heading back, a 3.5-day journey, before a critical-to-life condition was reached. Fortunately, this crew was still in cis-lunar space, meaning two-way light times are still short enough that communication delays with MCs on Earth were minimal. This allowed for the mitigation of the symptoms of this failure to be performed, in concert with the ground support team, quickly enough to get the crew home alive.

As Humans venture beyond cis-lunar space, two-way light times become ever longer. The time it takes for a communication to be sent from the vehicle to the Earth and back again increases with every kilometer traveled. As the vehicle approaches Mars, two-way light times can reach 40 minutes. This fact alone means that an Apollo 13 style failure response may be unfeasible in a truly life-threatening situation. To complicate things even further, every two years, the orbital paths of Mars and Earth bring them to opposite sides of the Sun. This conjunction event lasts for two weeks where the star completely blocks out communications between the two planets. If a safety-critical system were to fail while in Mars orbit, there may not be sufficient time for a telemetry from a sensor package to reach MCs on Earth, have them conclude that something is wrong, decide what is wrong and how to mitigate it, then have those mitigation instructions sent back to the vehicle for the Human crew to implement. In deep-space applications such as this, it becomes imperative from a safety perspective to have as much of this process contained locally onboard the vehicle, driving the need for a level of vehicle autonomy that has yet to be realized.

NASA has defined autonomy as having the ability to achieve defined goals while operating independently of external control (NASA 2015) (Fong, et al. 2018). Addressing the need for vehicle autonomy is a primary research goal of the NASA Space Technology Research Institute (STRI) Habitats Optimized for Missions of Exploration (HOME). HOME is a consortium of seven universities spread across the country in partnership with NASA and several relevant members of industry. HOME has been structured into five Research Thrusts (RT's). RT-1 is looking at vehicle functional design. The group is "providing context and defining a methodology for assessing autonomous and other emergent technology applications in a deep space habitat." (HOME 5-31-2022) In other words, RT-1 assures that the research being done by the rest of HOME is relevant in the context of deep space habitats and quantifies the value of technologies that arise from that research. They also guide the institution by formulating Design Reference Missions (DRM), contextual definitions, and success metrics.

RT-2 is researching vehicle self-awareness. HOME defines self-awareness as a vehicle's "understanding of its systems, the crew, and the external environment, including prior states, previous actions, and resulting consequences, to inform decision-making." (Klaus, et al. Rev September 30, 2020) RT-2 is furthering capabilities in this domain by developing a "fully integrated predictive and prescriptive analytics framework." (HOME 5-31-2022) This framework will facilitate autonomous Integrated System Health Management (ISHM) methods of complex and interdependent systems and prognostics capabilities for predicting RUL from sparse data and limited failure experiences. RT-2 also includes significant efforts in system modeling and causal inference to support Root Cause Analysis (RCA).

RT-3 is researching Human-Autonomy teaming. This group is working on incorporating “Human behavior, states, and desires into autonomous systems.” (HOME 5-31-2022) (Pischulti, et al. May 11, 2020) The goal is to make spacecraft autonomous decision-making more amendable to the complexities of the Human condition. This includes bi-directional communication between Humans and vehicle systems as well as robotic agents onboard. Bi-directional communication means that autonomous systems are learning from Human behavior and interpreting their intentions. It also means that autonomous decision-making is made explainable and understandable to Human crew and MC’s on the ground.

Self-Sufficiency is the domain of RT-4. HOME defines self-sufficiency as “The degree to which a vehicle can enable a mission to be accomplished within a specified increment of time, crewed or uncrewed, without external intervention.” (Klaus, et al. Rev September 30, 2020) This means autonomous management of logistics and resources along with robotic implementation of Maintenance and Repair (M&R) tasks. RT-4 is also looking at novel onboard additive manufacturing methods for in-situ production of replacement parts and associated processes such as part verification and classification.

As the aforementioned technologies and methods were beginning to take shape, it was realized that this vast body of work required a large variety of inputs from often disconnected sources. Input sources may include sensor data, simulated data, schematics, and inter-component (ontological) relationships, or historical information. It was envisioned that in order for these technologies to truly work as intended and raise the autonomy and safety of a deep-space habitat, there is a need for a local, integrated source for all of the required inputs. RT-5 was initiated to investigate the utility of a Digital Twin in providing the framework for such an integrating functionality.

This paper will introduce the concept of a Digital Twin and explore how it has been applied in terrestrial industries. Unique challenges of deep space human habitation will be discussed. Chapter 2 will outline specific use-cases relevant to human space flight. Chapter 3 will propose a Digital Twin Framework that could support the relevant capability needs and provide functional engineering requirements. Chapter 4 will look at a demonstration implementation and proof of concept.

Chapter 1: Introduction to Digital Twins

The Concept of a Digital Twin

The concept of a Digital Twin spans a gap between the concrete and the abstract. This duality contributes to the lack of consensus on a general definition. In the abstract, a Digital Twin can be thought of as a method or system of methods that relate the physical world with a digital representation of it. This relationship can be leveraged to benefit awareness, understanding, and control of the physical world. In the concrete, these methods can be tailored to accommodate a specific use-case, leading to a variety of definitions across industries. Despite the diversity in definitions, there are some generally accepted characteristics that are shown in Figure 2 and outlined below. The initial presentation of these characteristics is intentionally simplified. As the following chapters discuss the specific needs of deep space habitation, these characteristics will become less abstract. The requirements that come out of this work will guide a concrete definition and structure of a Digital Twin that is appropriate for supporting deep space vehicle autonomy.

General Anatomy

Digital Twins are commonly expressed as a framework which stem from an evolution of some more basic concepts. At the root there is always some Physical Asset under investigation. This could be any real system such as a building, a piece of machinery, or even a human being. A Digital Model, portrayed in Figure 1 a), is a digital representation of that Physical Asset at one static point in time (a snapshot). This is most often its as-designed or as-built condition but could also be a snapshot of the Physical Asset at the time the Digital Model was initiated. There is no data exchange between the digital and physical worlds such that the Digital Model remains static and is not updated as the state of the Physical Asset changes (FULLER, et al. 2020). The Digital Model may just be a 3D geometric model. It may also include kinematic or functional behavior models, physics and data-based simulation models, information about material properties, or component-specific characterization. If one values life history as an intrinsic part of a component's current state, then historical performance data may also be included.

If the Physical Asset is outfitted with some means of collecting information about it, then this information can be passed to the digital representation, thus creating a Digital Shadow as shown in Figure 1 b). This usually means sensor data but may also include human observational data. The Digital Shadow is continuously updated to reflect any change in state of the Physical Asset. The fidelity of a Digital Shadow should be such that any information that could be

obtained from inspecting a Physical Asset can also be obtained from its Digital Shadow with comfortable uncertainty. This data exchange only happens in one direction, there is no information or influence sent to the physical world (Baidya, et al. 2022).

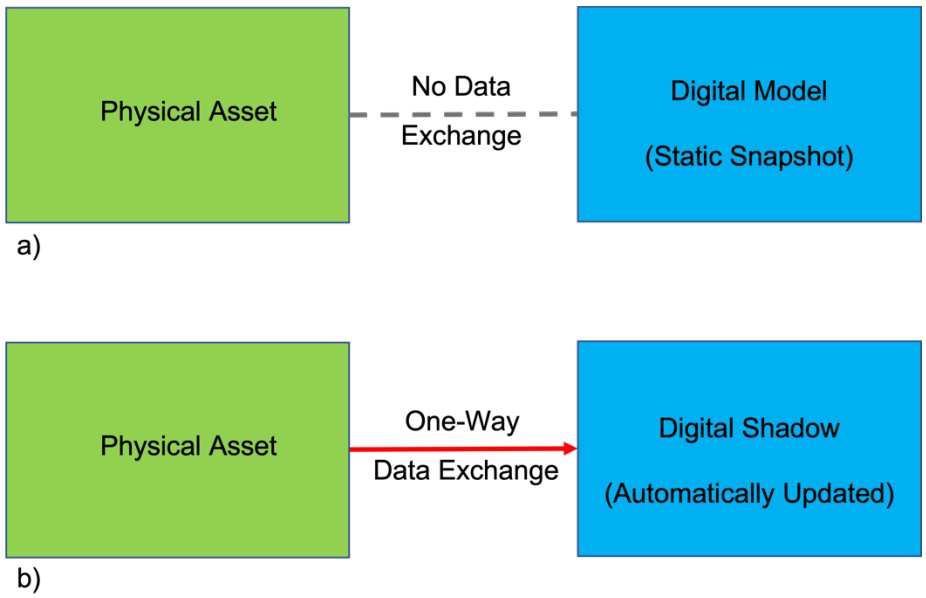

Figure 1: a) Representation of a Digital Model. Showing no information interaction between the Digital Model and the Physical Asset. b) Representation of a Digital Shadow. Showing the one-way Digital Thread connection with the Physical Asset.

Of course, there is little point in understanding the current state of a physical system unless that knowledge is used to benefit the physical asset in some way. A Digital Twin Framework is established by closing the loop between the digital and physical worlds. A bi-directional exchange of information and influence between a Physical Twin and a Digital Twin constitutes a Digital Twin Framework. Figure 2 depicts an abstraction of this framework followed by a brief description of its primary components (Baidya, et al. 2022).

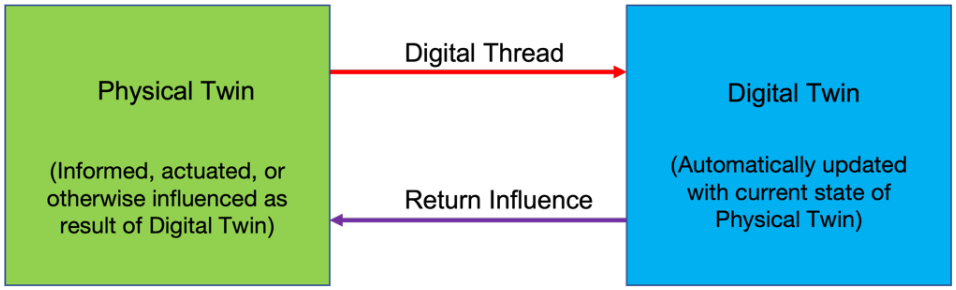

Figure 2: Simplified Digital Twin Framework. General anatomy is intentionally depicted as abstract. This framework will become more developed throughout this paper.

Physical Twin

The Physical Twin is some real system under investigation. This could be any real system such as a building, a piece of machinery, or even a human being. The Physical Twin is outfitted with some means of collecting information about it. Most often this means sensors but may also include human observational data or expert analysis. It is, in fact, the Physical Asset mentioned above in a reciprocal relationship with a Digital Twin whereby there is a bi-directional exchange of information and influence between the two.

Digital Twin

The Digital Twin is a virtual representation of the Physical Twin that is automatically updated as to the current state of the Physical Twin. The Digital Twin commonly includes a high-fidelity geometric model, information about material properties, operational parameters, or historical performance data, and some ontological model of how all the components of the physical system are related. The ultimate value of a Digital Twin comes from the fact that more information can be gotten from the virtual entity than from observation of the physical entity, or a net information gain. This statement may feel like a stretch, but it's meant in terms of a practical application rather than a theoretical application. It is true that any value that could be simulated could conceivably be measured. Otherwise, the model used in simulation could never be validated. However, real engineering systems cannot generally accommodate infinite sensors. There will always be state values which a real engineering system is not outfitted to measure in practical use. The Digital Twin can also be used to obtain information about past states and predict future states, which mere observation of the Physical Twin may not be able to provide.

Digital Thread

The Digital Thread is the digital information pathway from the Physical Twin to the Digital Twin. In the case of sensor data, the Digital Thread may consist of a data acquisition (DAQ) device, some IoT messaging protocol like MQTT (Vering, et al. 2019), and a software package that takes the information coming from the DAQ and updates the Digital Twin appropriately. In the case of human observational inputs, the Digital Thread may manifest as a human-software interface.

Return Influence

An important characteristic of the Digital Twin framework is the Return Influence which passes some beneficial influence from the Digital Twin to the Physical Twin. The Return Influence can

be a digital controller, informed by the Digital Twin, that sends a control signal back to a plant or delivers some other instructions to the physical asset. It may also manifest as a human-in-the-loop task. For example, a repair crew may be signaled to perform condition-based maintenance on a piece of machinery or aircraft structural element.

Applications Across Industry

The following literature review includes applications for Digital Twins across four industries: Architecture, Engineering and Construction/Facility Management (AEC/FM), Manufacturing, Medicine and Healthcare, and Aeronautics. This review is given to create a foundation to build on and a tool chest of established Digital Twin characteristics to use in the development of a Digital Twin Framework that could support an autonomous deep space habitat.

Architecture, Engineering and Construction/Facility Management (AEC/FM)

(Xie, et al. 2020) proposes a Digital Twin framework that combines Building Information Model (BIM) and data driven asset monitoring system to enable automated condition monitoring and anomaly detection. The authors envision a Facilities Management (FM) operation that would see essentially no unexpected failures of critical systems.

(Vering, et al. 2019) describes a process for developing a Digital Twin to support HVAC product lifecycle management. The study builds upon models developed during concept and design phases. Bayesian model calibration is conducted in transition to the as-built state using physical twin measurement data. The authors successfully demonstrated that implementation of the developed Digital Twin improved system energy efficiency and predictive maintenance for the operating HVAC in a building.

According to (Sacks, et al. 2020), a major challenge with creating an efficient Digital Twin for large construction projects is the inherent participation of many independent contractors, suppliers, designers, etc. Each of these participating entities may use any number of digital tools and data formats that may not be immediately compatible. Like other literature in this field, the authors also talk about the potential utility of a well-informed what-if scenario assessment. They propose that the ability to predict how a project will unfold given certain future parameters can help guide production planning and reduce construction waste, as is the goal with lean construction.

Innovation in IoT is driving the increased connectivity of personal devices carried and used by people. This expanding connectivity is also allowing for a large number of sensors to be outfitted on municipal assets and city infrastructure of population centers. A Digital Twin can allow for the integration of all these IoT connected devices and sensors to create valuable tools available to city planners, municipal managers, and service providers. Digital Twin enabled tools can benefit city planning, public safety, utilities distribution, energy usage, and transportation efficiency. (Mohammadi and Taylor 2017) proposes that a Digital Twin of a city can simulate what-if scenarios regardless of the current state of the city. The author suggests that this gives city planners the ability to forecast how the state of the city would evolve due to the adjustment of various parameters such as economic, environmental, and social changes.

Manufacturing

(Huang, Wang and Yan 2020) proposed a Digital Twin for use in the design of Reconfigurable Machine Tools (RMT). Experience and performance from existing RMTs is tracked through a Digital Twin to drive the design of new RMTs. To access the abundance of configurations possible with RMT modular design, the study argues that a Digital Twin should contain libraries of module models and ontologies that cover all possible reconfigurations. Thus, the Digital Twin should be aware of all possible reconfigurations of the tool. Missing from this study is a discussion of how such a reconfigurable Digital Twin could be used during RMT operation.

(Shao and Helu 2020) prescribe a Digital Twin framework to be standardized across the manufacturing industry to improve decision making and process control. The three use-cases investigated are: minimizing the impact of equipment downtime, optimizing production planning and scheduling, and enabling virtual commissioning. The proposed digital twin utilizes live data acquisition, real-time state analysis, and fault prediction and diagnostics algorithms.

Medicine and Healthcare

Driven by the SARS-COV-2 pandemic and the very broad range of patient reactions to the virus and treatments, (Laubenbacher, et al. 2022) outlines a high-level roadmap to the realization of a Digital Twin of the Human immune system. The study suggests that such a Digital Twin could have far reaching benefits in patient-specific diagnosis, prognosis, and therapy optimization. One example use case is the targeting of a sufficient immune response to a pathogen while minimizing inflammatory tissue damage. The authors suggest a hierarchical modeling structure whereby various levels (e.g., molecular level, cellular level, tissue level, organ level, and body level) are integrated into a comprehensive multiscale base model.

An extensive literature review by (BJELLAND, et al. 2022) investigated how cutting-edge modeling and computational techniques can be used in a Digital Twin architecture to support arthroscopic knee surgery, specifically targeting patient-specific preoperative planning and resident doctor training. The study recommended high-fidelity modeling and simulation with patient-specific model parameter calibration integrated with expert knowledge and patient records. They also promoted a sophisticated user interface with haptic feedback for surgical training.

Aeronautics

(Bellinger, et al. 2011) argues that the automation of numerical modeling has advanced along with the understanding of underlying physics of aircraft structural fatigue and failure. However, the process that engineering groups go through to estimate structural fatigue has not moved much in the past 60 years. This lack of evolution in engineering processes has led to programmatic issues in the design of revolutionary aircraft, e.g., weight, budget, and schedule overruns. (Bellinger, et al. 2011) recommends two ultra-high-fidelity Digital Twin's specific to each aircraft tail number. Both Digital Twins are comprised of an ultrarealistic geometric model. One Digital Twin is then coupled with a conglomerate of physics-based simulation models. The other Digital Twin acts as a repository for sensor data from the physical system. The simulation Digital Twin would take probabilistic operational parameters as input and predict crack nucleation and propagation for any given planned flight of the aircraft. As the aircraft is operated, real sensor data is recorded in the sensor Digital Twin which is used to update the simulation Digital Twin using Bayesian methods. This process is used to forecast remaining useful life and updating reliability estimates for critical components.

The role that Digital Twins have historically played in aeronautics have been primarily focused on monitoring structural fatigue in aircraft due to the cyclical loading of flight operations. This process always happens offline and on the ground. (Liao, Renaud and Bombardier 2020) combined this idea with a fleet-wide database that updates individual, tail number-specific Digital Twins with probabilistic usage. The goal is to predict the RUL of the aircraft structure and improve condition-based maintenance. The study is intended for U.S. Air Force fleet management to increase safety and reduce the lifecycle cost of each tail number in the fleet.

The previous two studies discussed integrated operational data into a Digital Twin offline and after a flight was completed in order to make maintenance or operational decisions regarding the next flight. (Kapteyn 2021) demonstrated that a Digital Twin could be continuously updated during flight and actively utilized to create a self-aware UAV, improving decision making throughout a UAV's operational life. Kapteyn proposed a mathematical and computational

framework, based on a probabilistic graphical model, that defines how a Digital Twin interacts with and evolves with and its associated Physical Twin and allows for scalability and flexibility of use. The study made use of scalable reduced-order models to accelerate the processing of structural fatigue models in order to make that increased knowledge and awareness useful during flight.

(Guivarch, et al. 2019) proposes a Digital Twin to support prediction of mechanical part lifetime. This becomes especially useful where instrumentation is difficult or not possible to implement, as is the case with helicopter rotor systems. To fill in the knowledge gaps due to limited sensors, this study implements simulation generated data. The Digital Twin takes real data from adjacent, more easily sensored components and inputs that data into simulation models to produce state estimates of components that cannot be sensored, effectively creating a virtual sensor.

Summary of Industry Capabilities

Table 1 presents a summary of the Digital Twin applications reviewed above. The specific use-case is given for each application along with the Digital Twin characteristics that support those use-cases.

Table 1: Summary of Digital Twin applications across industries.

Industry	Use-cases	Supporting DT Characteristics	Reference
AEC/FM	Automated condition monitoring and anomaly detection.	Integrated information models	(Xie, et al. 2020)
AEC/FM	Production planning, waste reduction	Integration of disparate models and information sources, What-If engine	(Vering, et al. 2019)
AEC/FM	Improved energy efficiency and predictive maintenance, decreased product life cycle cost	Model calibration, IoT integration	(Sacks, et al. 2020)
AEC/FM	City planning, public safety, utilities distribution, energy usage, and transportation efficiency	IoT integration of sensors and data sources, What-If engine	(Mohammadi and Taylor 2017)

Manufacturing	Reconfigurable tool design	Libraries of models and ontologies to cover all possible reconfigurations	(Huang, Wang and Yan 2020)
Manufacturing	Minimizing equipment downtime, optimized planning and scheduling, virtual commissioning	Industry standardized framework, live data acquisition, state analysis, fault prediction and diagnostics algorithms	(Shao and Helu 2020)
Med & Health	Patient-specific diagnosis, prognosis, and therapy optimization	Hierarchical modeling structure, System-of-systems modeling	(Laubenbacher, et al. 2022)
Med & Health	Patient-specific preoperative planning, resident doctor training	User interface with haptic feedback, high-fidelity modeling and simulation, integration of expert knowledge and patient records, model parameter calibration	(BJELLAND, et al. 2022)
Aeronautics	Remaining useful life and updated reliability estimates for critical components	Ultrarealistic geometric model, integrated physics-based simulation models, sensor data repository, Bayesian model calibration	(Bellinger, et al. 2011)
Aeronautics	Large fleet management, condition-based maintenance	Multi-asset federated Digital Twins, Bayesian model updating	(Liao, Renaud and Bombardier 2020)
Aeronautics	Self-aware UAV, mid-operation health-aware decision making	Rigorous mathematical framework, probabilistic graphical models, scalable reduced-order models	(Kapteyn 2021)
Aeronautics	Virtual sensors for state estimates of difficult-to-sensor components	Integration of live sensor data with advanced multibody simulations	(Guivarch, et al. 2019)

Considerations for Deep Space Habitation

As of this writing, very little work has been done in the implementation of Digital Twin technology in the field of deep space habitation. What is needed for spacecraft meant to support Human life far from Earth is a robust Digital Twin ecosystem that can enable heightened levels of autonomy during the operational phase of the spacecraft's lifetime. That is, a Digital Twin that can support the critical functions of the spacecraft from deployment until end of life with a particular emphasis on supporting the day-to-day functionality of the spacecraft. This need was first recognized by NASA in 2012 and outlined in (Shafto, et al. 2012). The HOME institute has, as of this writing, not been able to find a comparable example of such a Digital Twin deployment on a human crewed spacecraft. The novelty of this concept and the potential impact to NASA's broader interests in ensuring safe and sustainable Human exploration is the driving force behind this work. Even so, there have been notable efforts to develop adjacent and supporting technologies which will be discussed in this section.

Human rated spacecraft are immensely complex vehicles. They contain systems and often-redundant subsystems that are critical to Human life as well as coupled in numerous ways. To fully understand the evolving state of the entire vehicle, all the complex dependencies between systems need to be understood, modeled, and continually updated. This would require significant onboard computational resources. On Earth, it may not be too much of a stretch to accumulate the necessary resources into a single computer. Afterall, computers can be scaled essentially without limit given enough space and power. In space that is not the case for three reasons: mass and power restrictions, system redundancy, and transistor density.

Driven primarily by the energy costs of leaving Earth's gravity well, space missions are designed with tight constraints on volume, mass, and power. So super computers are out of the question. Even if a very heavy, power-hungry computer were designed into the vehicle, engineers could never guarantee that a computer would never fail. For this reason, space missions are always designed to accommodate multiple redundant computer systems. One may reason that transistor densities are ever increasing and that a significantly powerful processor may enable smaller computers. This reasoning also becomes problematic as the vehicle enters deep space, leaving the protection of Earth's magnetic field behind. Beyond LEO, high-density processors greatly increase the chance that a transistor may be struck by radiation from the Sun or gamma rays passing through our star system. Decentralizing the computational load with a distributed, hierarchical architecture appears to be a better solution for avoiding these radiation-caused "Single Event Upsets".

A 2019 study led by Dr. Julia Badgers at NASA Johnson Space Center (Badger, Strawser and Claunch 2019) proposed a component-based system that begins to address distributed processing while providing interfaces for developing autonomous technologies. The proposed Modular Autonomous Systems Technology (MAST) framework is divided into "buckets" of autonomous functionality that exemplify the OODA loop (Observe, Orient, Decide, Act) concept. A complete set of these autonomous functionalities are a "cluster". A spacecraft is then divided into a hierarchical structure. An example of such a structure might be vehicle level (e.g., Vehicle System Manager), element level (e.g., Habitat System Manager), system/process level (e.g., Environmental Control and Life Support), and subsystem level (e.g., Water Recovery System). Each component within the structure has its own cluster responsible for the autonomy of that element constituting a distributed system. This allows the delegation of tasks to move up and down the hierarchical structure. Telemetry and processing requests can move up to higher levels of the structure, for example, while commands and telemetry requests can move down to lower levels.

The MAST conceptual framework opens the door for much more manageable, reduced-order models to be used in simulation rather than expensive, high-order system-of-system simulations. Vehicle complexity presents another issue with autonomy itself: just as it is difficult to model all possible system interactions, it is also difficult to anticipate and plan for all possible scenarios that an autonomous software would encounter. Traditional brute force autonomy depends on hard-coded solutions to specific applications. Because of this, truly autonomous systems have been difficult and expensive to develop and deploy successfully. The natural limits in Human forethought have been a limiting factor in the scope and effectiveness of autonomous systems.

An effort out of NASA Stennis Space Center's Autonomous Systems Laboratory (ASL) suggests that "thinking" autonomy is dynamic and application agnostic (Figueroa, Underwood and Walker, et al. 2019). ASL has been developing a complete software platform that supports real time "thinking" autonomy, based on generic first principle models, that promises to bring down the cost of implementing the level of autonomy needed to realize NASA's spacecraft operations support goals. The resulting NASA Platform for Autonomous Systems (NPAS) software platform extends Model-Based Systems Engineering (MBSE) to incorporate live models which can be updated and evoked in real-time. NPAS supports autonomy methods that can be more affordably implemented and are able to reason independently from ground control.

Another notable effort out of NASA Ames Research Center seeks to address difficulties in anomaly detection and response that arise due to deep-space communications issues. As previously discussed, heightened two-way communication latency and the potential for

complete comms black-out during conjunction events necessitates that greater agency be given to crew and local systems onboard the spacecraft to deal with off-nominal events. The Advanced Caution and Warning System (ACAWS) was developed as an Integrated System Health Management (ISHM) package to support this need. ACAWS consists of four major modules: anomaly detection, fault detection and isolation, system effects analysis, and a Graphical User Interface (GUI) connected through a data distribution middleware (Colombano, et al. 2013). The software package offers real-time ISHM mission support and promotes greater cross-collaboration between MC's and crew. Displays, interfaces, and level of interaction with the system can be adjusted depending on the operator. ISHM plays a critical role throughout this work. See (Xu and Xu 2017) for a good introduction to this topic.

In step with many NASA Mars mission architectures, the HOME DRM envisions an orbital habitat that is assumed to be unoccupied 80% of the time over its 15-year operational lifespan (Pischulti, et al. May 11, 2020). This means the habitat spends a significant amount of time with no crew present, often referred to as dormancy. The United States has little experience with Human-rated spacecraft dormancy with only Skylab having short dormant periods. The length of dormancy anticipated for a Mars orbital habitat is well beyond precedence and will require an operational paradigm shift from the state-of-the-art. (Badger and Frank 2018) conducted a functional analysis of subsystem operation during dormancy. The study found that despite the absence of crew, nearly all spacecraft subsystems will continue to be active at full or near full nominal capacity. The exception is the Environmental Control and Life Support System (ECLSS) as there is no longer a reason to maintain an environment that is hospitable to Humans. Certain functions will still be required however, to maintain safe functionality of electronic systems onboard, the Temperature and Humidity Control System (THCS) for example. The capacity of the Carbon Dioxide Removal System (CDRS) will likely be greatly reduced or may be powered down completely.

CDRS is among the most essential systems for supporting Human life onboard any space habitat. Loss of CDRS operation is one of the quickest ways to a loss of crew scenario. Therefore, it is essential that this system be brought online to full capacity and thoroughly tested before crew arrival. As is commonly observed with many mechanical systems on Earth, long periods of dormancy can influence things like material and adhesion properties (e.g., O-ring dry rot or cold welding between metal surfaces) and may affect the way a system operates when it is brought back online. Digital Twin technology has the potential to be a tool in the assessment of any faulty components or deviation in the operational baseline of a system "waking up" from dormancy.

If the SmartHab Digital Twin-enhanced ISHM system determines there is a safety-critical issue requiring resolution prior to crew arrival, the habitat will need to be outfitted with autonomous robotics platforms capable of performing any M&R tasks required. In fact, robotic M&R may be necessary during any phase of flight operations. Issues due to normal wear and tear during dormancy will need to be addressed with robotic systems. There may also be M&R tasks that robotics are simply better at or more suited to handle than their Human counterparts during crewed phases of operation. A Digital Twin should be able to assist in robotic autonomy by supporting volumetric spatial awareness through high fidelity and updatable geometric models of the habitat. A Digital Twin should also be able to assist in robotic task planning and logistics tracking.

It is clear that robust “thinking” autonomy will play a critical role in the operation of the spacecraft and the maintenance of its systems through software and robotics packages. Each phase of flight; crewed, uncrewed, and any transition periods between the two have their own system requirements, challenges, and available resources. Therefore, the level of autonomy that is needed for uncrewed phases may not be appropriate for crewed phases and vice versa. The exact proportions of vehicle control that is given to crew, MCs, and the autonomous systems of the spacecraft will vary depending on the phase of the mission and will also evolve as experience is gained over the lifetime of the SmartHab. This suggests the notion of variable or adaptive autonomy. Even though the methods of adaptive autonomy are beyond the scope of this work, it is important to assure that any Digital Twin framework has the flexibility built into its architecture to accommodate for this important concept.

Chapter 2: Deep Space Habitat Relevant Use-Cases

The spectrum in which Digital Twin technologies can be used to support space vehicle autonomy and make deep-space Human habitation safer, is extensive. When considering the complexities of deep-space habitat operations such as are described in the HOME DRM, one could generate a great number of scenarios in which a Digital Twin might be useful. In the interest of scope, this work will target a set of seven specific use-cases that would maximize the return on research investment of having an onboard Digital Twin capability within HOME. We also address the boundaries of current DT capabilities with respect to deep-space habitats, and promote a robust Digital Twin ecosystem with the highest benefit to spacecraft operations and crew safety. These use-cases were developed to represent a cross-section of capabilities thought to be made possible or better by the implementation of a Digital Twin. The use-cases selected for investigation are:

- System Degradation Estimation and Prognostics
- Root Cause Analysis
- Cross-System Awareness and Explainability
- System Checkout During Prep for Crew Arrival
- Safety-Critical What-If Questions Without Risk to Hardware
- Global Map Keeping
- Logistics Tracking

The HOME Research Thrust 1 (Vehicle Functional Definition) Concept of Operations for the SmartHab has defined four operational states for a deep-space habitat: habitable, uninhabited, uninhabitable, and degraded. (Pischulti, et al. May 11, 2020) The DRM defines habitable as a vehicle in which a Human crew is present, and the state of the habitat is such that it is safe for them to be there. Uninhabited means that a Human crew is not present, but the vehicle is maintained in a safe condition. Uninhabitable means that a non-survivable failure exists, and the vehicle is not safe for Humans. Degraded is a condition in which a survivable failure mode exists; the vehicle is still safe for a Human crew if timely corrective action is performed.

Figure 3 shows how the use-cases under investigation map to these four operational scenarios. A full description of each use-case follows. There are several software modules that appear throughout the described use-cases; however, it should be noted that they are only mentioned

as participating agents and the details of their internal functions and algorithms are outside the scope of this work.

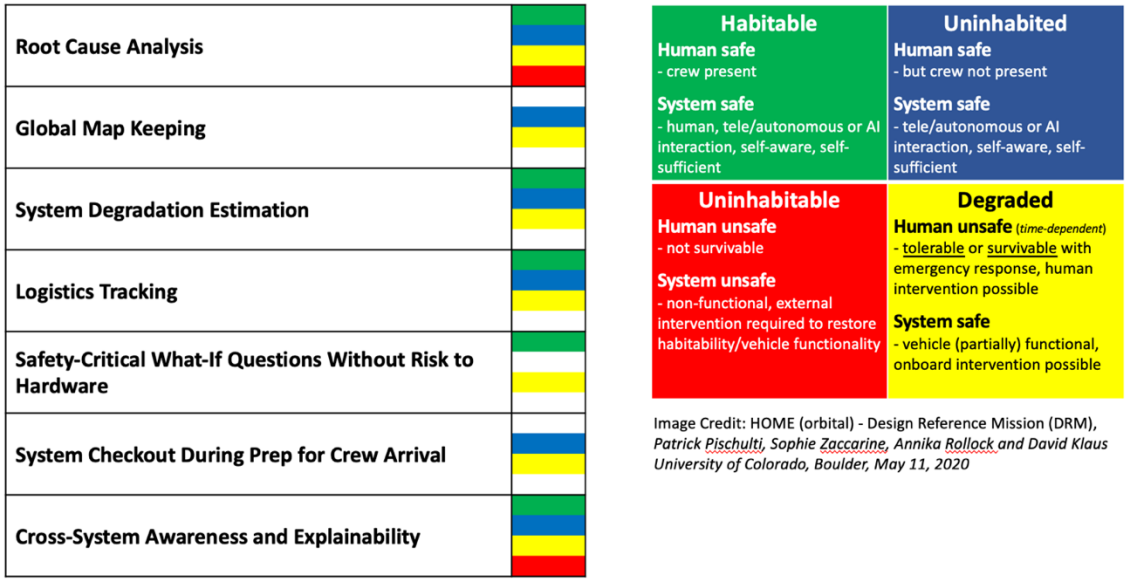

Figure 3: Mapping of use-cases to the four operational scenarios outlined in the HOME DRM.

Reference Vehicle Architecture

For clarity, a brief discussion of the envisioned deep space habitat and all its major components is warranted. The reference vehicle featured in this work is a hypothetical human-rated spacecraft in orbit around Mars called the Integrated Orbital Habitat (IOH) (Fig 4). The IOH is created by the docking of the Mars Transfer Vehicle (MTV) and the Mars Orbital Habitat (MOH).

The MTV serves to transport crew from the Earth System to the Mars System and back. Its major components are the Habitat Module, Logistics Module, and Orion Spacecraft. The Habitat Module hosts the primary living quarters for the crew, exercise equipment, and a multi-purpose laboratory space. The logistics module is storage for all of the consumable resources for the entire mission as well as spare parts and resupply manifest for the MOH. The Orion Spacecraft transports crew between Earth's surface and the MTV. It also serves as the primary propulsion unit for the MTV.

The MOH remains in orbit around Mars for its entire operational lifetime. It operates in a dormant state when the MTV is not docked (crew is not present). The major components of the MOH are the Docking Node, Laboratory Module, Power & Propulsion Element, and Mars Surface Lander. Figure 4 depicts a high-level diagram of the discussed reference vehicle and all of its major components.

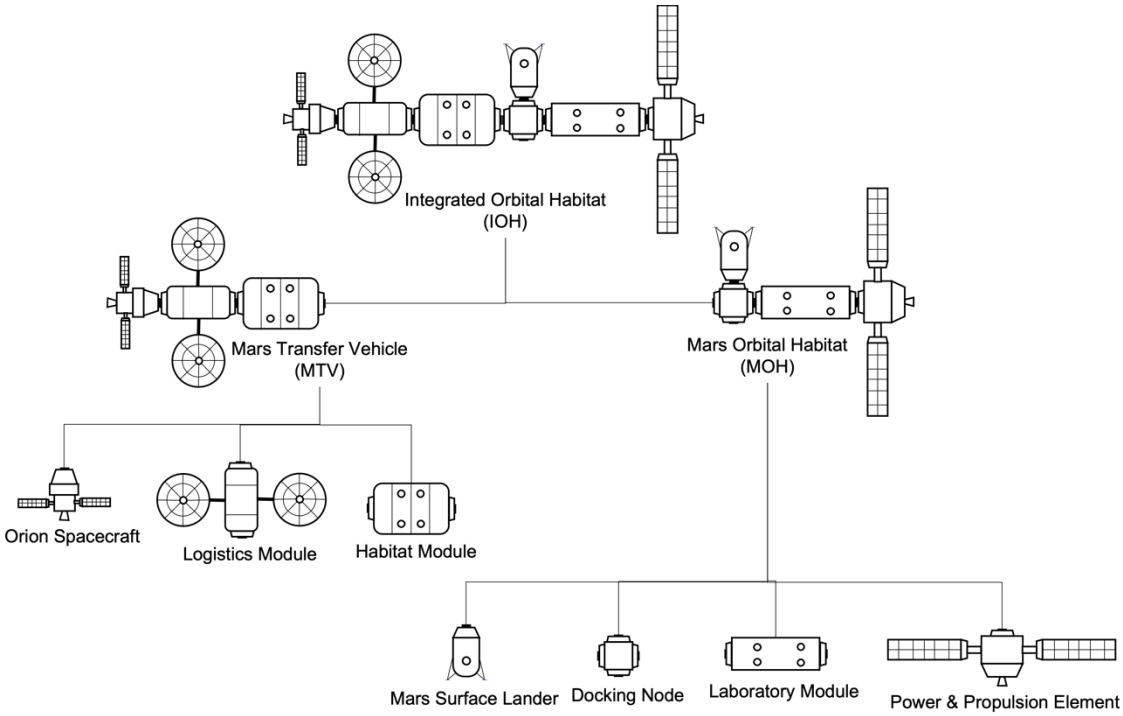

Figure 4: High-level diagram showing the reference vehicle for this work and all its major components.

System State Estimation

Motivation and Overview

System state estimation is a critical first step in enabling many capabilities necessary to complete the autonomy puzzle. Before one can begin to diagnose anomalies, understand the root cause of an issue, prognosticate a remaining useful life, or predict future behaviors, one must first understand the present state of the system under investigation. One could look at sensor data for this insight, but how can you trust a sensor is working properly or has not drifted from its calibration point? State estimation methods can act as a check, giving an expected value to be weighed against real data coming in from a sensor package.

More importantly to the context of spacecraft, is the mass and complexity penalty that comes with extensive sensor inclusion. Aside from the physical structure, cable harnessing is among the most massive aspects of any spacecraft. If engineers were to use a sensor to measure every piece of information necessary for a truly wholistic understanding of the state of a spacecraft, the mass penalty due to cable harnessing would quickly become prohibitive. There are also instances where it is not practically feasible to place a sensor without impacting system performance or due to mechanical constraints. Also, sensors can fail, and in-mission replacement may be impractical. Vehicle state estimation methods can use real, causally

adjacent sensor data to generate virtual sensor data. This may reduce the number of sensors needed to track the health of critical spacecraft components or fill in data gaps where sensors cannot be used. This topic will be explored further in Chapter 4.

Critical systems in a human-rated spacecraft include the Electrical Power System (EPS) and the Environmental Control and Life Support System (ECLSS). The ECLSS is the primary system responsible for keeping humans alive. The importance of understanding the state of health of ECLSS subsystems such as the Atmosphere Revitalization System (ARS) and the Water Recovery System (WRS) cannot be understated. It may be counterintuitive, but the most commonly replaced components within these systems and their failure modes are not necessarily exotic or overly complex. Table 2 provides a few examples of key systems and typical failure modes based on the author's experience investigating and databasing International Space Station failure modes at NASA Johnson Space Center.

Table 2: Typical failure modes of three critical ISS life support systems.

Atmosphere Revitalization System (ARS)	Water Recovery System (WRS)	Electrical Power System (EPS)

Prone to:
• Filter clogging
• Valve stickage
• Zeolite dustification
	Prone to:
	• Mechanical failure
	  • Pumps
	  • Belts
	  • Bearings
	  • O-rings
	• Filter breakthrough
		Prone to:
		• Battery degradation
		• Vibrations
		• Micro meteor impact and elemental oxygen degradation of SA's

In a Terrestrial context, these failure modes do not seem overly daunting. A quick run to a hardware store and the turn of a wrench could have a system up and running quickly. The problem is that there are no hardware stores in deep space. Lead times for resupply are mostly a function of orbital mechanics and can range from approximately 8 to 34 months for a Mars outpost. The higher value assumes the Earth-Mars transfer window was missed. In such an extreme high-risk and remote environment, any of these failures can quickly lead to a loss of mission or even loss of life scenario. Having the ability to track and log present and past states and estimate the future states of the above-mentioned systems is crucial. Additionally, being able to prognosticate future system states with good confidence means that the failures listed in Table 2, and many others, could be mitigated before they even happen. Essentially, there should be no surprise failures in any critical system.

A Digital Twin can accommodate this process by acting as a central repository for sensor data, operational parameters, information about form and configuration, and more. Containing all of these disparate data in a centralized location increases the ease and efficiency of access by algorithms supporting fault detection, diagnostics, and prognostics. A Digital Twin can also act

as an integrator of models. Physics and data-based models, contained within the Digital Twin, can be called upon by the same algorithms. Centralization within the Digital Twin enforces across-the-board conformity across models and information sources and provides for verification of information.

In the use-case description below, the example of an atmospheric Carbon Dioxide Removal System (CDRS) sorbent bed is used. This use-case will be greatly expanded upon in Chapter 4. A sorbent bed creates an interesting case study because it is not possible to outfit a sensor package inside the sorbent bed without affecting its performance. Therefore, it is an ideal candidate for a virtual sensor as described above. In this example, sensor data is collected (airstream pressure, temperature, relative humidity) upstream and downstream of the sorbent bed and stored in a data repository. State estimation modules should be able to access information stored in the Digital Twin as well as request physics model simulation results to compute an estimate of the current state of the interior of the bed. Results of model estimates should be accessible and interpretable to root cause analysis algorithms and preventative maintenance scheduling protocols.

Detailed Use-Case Description

Use Case	State Estimation of CO_2 Removal System (CDRS) sorbent bed
Subject Area	System degradation estimation/Prognostics/Preventative maintenance
Primary Actor(s)	State Estimation Module (software)
Precondition 1	There is a zeolite-based sorbent bed component to the CDRS of the Air Revitalization System (ARS).
Precondition 2	The system is turned on and operating at full capacity.
Precondition 3	The Digital Twin has knowledge of the current configuration of the ARS, including its components, their attributes, their sensors, etc.
Precondition 4	The required information for performing state estimation is available. Sensor data available is of adequate type, quantity, and quality with known uncertainty.
Precondition 5	The Digital Twin has sufficient physics-based models to describe the functioning of the sorbent bed.
Triggers	Runs at some pre-determined sample rate.
Basic Flow	
Description	The system runs at some sampling rate, monitoring sensor data. States are estimated for the interior of the sorbent bed for use by Integrated System Health Management processes.

Observe	1	Embedded sensor package monitors inlet/exit sorbent bed parameters (Pressure, temperature, RH, ...) with some sufficient sensor placement configuration.
	2	Sensors provide their readings through some DAQ and interface.
	3	Sensor data is sent to a dynamic repository in the Digital Twin via some software interface.
Orient	4	State Estimation Module asks the Digital Twin for information about the ARS and its components, along with sensors in it.
	5	State Estimation Module selects the sensors that it can use to estimate status of the sorbent bed and asks for data.
	6	The Digital Twin interface produces the logs and hands them over to the module.
	7	State Estimation Module requests physics-based model results based on model parameters it provides to the Digital Twin.
	8	The Digital Twin processes the simulation model as requested and returns the results to the State Estimation Module.
Decide	9	The State Estimation Module then analyzes the data using its internal logic and produces an estimate of the current states-of-interest of the CO_2 removal system sorbent bed.
Act	10	The current state estimates of the sorbent bed are communicated to the Digital Twin and stored, along with metadata about these estimates (e.g., which agent/version produced them, what uncertainty was communicated, the date/time of the estimate, etc.).

Root Cause Analysis

Motivation and Overview

Traditional methods of Root Cause Analysis (RCA) depend on manual processes that produce largely static models like fault trees and dependency matrices. These models are developed by system experts based on both prior experience with similar systems and educated reasoning at the time of system development. The models are applied with the assumption that they are all inclusive and that fault-symptom relationships remain constant over time. The problems with these assumptions begin with the highly coupled nature of spacecraft systems. Systems that would not normally be causally adjacent in terrestrial applications begin to have influence on each other when packaged together on a spacecraft. An example is the Atmosphere Revitalization System (ARS) and Water Recovery System (WRS) introduced in the System State Estimation use-case. For most buildings, the water system and air system, commonly referred

to as HVAC, are open systems. That is, they have a fairly free exchange of energy and mass with the world outside the building. The influence they impart on each other is negligible compared to the influence from the outside world. When those same systems become closed systems onboard a space habitat though, their mutual influence can become non-negligible. To say that an engineer, no matter how informed, is going to be able to predict every way in which these systems could interact is a bit of a fool's errand, especially when taken in the context of a deep space habitat where human lives are at stake.

All this drives the necessity for autonomous and continuous fault-symptom relationship updating on a deep-space habitat. An effort within NASA HOME STRI led by Min Hwang of Carnegie Mellon University is seeking to address this issue and develop methods for an adaptive Root Cause Analysis (RCA). There may also be future reconfigurations to the habitat that were not foreseen when the system and associated fault-symptom analysis was developed. It is possible that the expansion of the habitat with additional modules or visiting vehicles may introduce unplanned redundant systems. In another HOME study, Hwang showed that interconnected redundant systems can have a profound effect on fault causality (Hwang, Akinci and Berges 2022). Another complication arises when considering that testing conditions on the ground cannot precisely imitate the actual operational environment of deep space. Because of this, it may not be possible to flesh out all causal relationships before system deployment. Given these complications, it is clear that traditional RCA methods must be augmented to allow for causal relationships to be updated dynamically throughout the operational lifetime of the habitat.

A Digital Twin can assist such an RCA module by containing a global Fault-Symptom Relationship Model (FSRM). Fault-Symptom Relationship Models can be expressed in several forms as discussed in (Hwang, Akinci and Berges 2022). The global FSRM is a complex system-of-systems model consisting of subsystem FSRMs federated in such a way that shows how causally adjacent systems affect each other. The FSRM should be fully updatable by external modules, although the algorithms behind that update are beyond the scope of this work. The Digital Twin should also contain and provide easy access to semantic information models, current sensor data, and historical nominal telemetry data.

The following use-case describes a situation in which the inaugural crew has arrived at the habitat. The Mars Transit Vehicle (MTV) that the crew arrived on has its own Temperature and Humidity Control System (THCS). Once the MTV has docked to the habitat and the hatch opened, they now represent an integrated volume of atmosphere serviced by two redundant THC systems. This reconfiguration was planned, and engineers did their best to develop a FSRM that accounts for the redundant interconnected system. However, this is the first time the

interconnected system has been operational in its intended environment, and it is likely that the FSRM will need to be updated to reflect the true nature of this reconfiguration. In this scenario, a crew member initiates this update, however it may also be initiated by Mission Control or autonomously on the spacecraft. Additionally, a fault has been detected triggering the RCA Module to assess the situation.

Detailed Use-Case Description

Use Case	Root Cause Analysis
Subject Area	Fault Management
Primary Actor(s)	RCA Module, Fault Detection Module
Precondition 1	The Digital Twin has knowledge of the current configuration of the THCS, including the components in it, their attributes, the sensors in them, historical observation data, and control logic of the system.
Precondition 2	The Digital Twin contains a causal model that depicts all fault-symptom relationships to the extent of current knowledge and preparation.
Precondition 3	The Digital Twin also contains a causal model of the anticipated combined vehicle consisting of the habitat and transfer vehicle that was sent as a data package from Earth ahead of MTV arrival.
Precondition 4	Docking of the transfer vehicle to the habitat was successful and no issues with the connection between the two vehicles has been detected.
Precondition 5	The THCS on both vehicles are operating redundantly.
Triggers	FSRM update triggered by crew member. RCA triggered by the exceedance of a predetermined sensor value threshold or other fault detection algorithm.
Basic Flow	
Description	The first crew has arrived at the habitat. The Mars Transit Vehicle (MTV) that they arrived on has docked with the habitat creating an integrated habitable volume. The MTV has brought its own THCS, which is kept in redundant operation to better service the expanded internal volume of atmosphere. Sometime later, a fault is detected, triggering the RCA Module to assess what happened.
Observe	1 Following the integration of the MTV and habitat, a crew member onboard the integrated vehicle accesses a Digital Twin-human interface screen and indicates to the Digital Twin that the previously planned reconfiguration has occurred, and that the system should update the FSRM appropriately.
Orient	2 Anticipating this reconfiguration, the Digital Twin already contains a FSRM of the integrated system. The Digital Twin switches its primary causal model to this new FSRM.

Decide	3	The Digital Twin triggers the RCA Module to update the FSRM of the integrated THCS to account for unforeseen causal relationships.
Act	4	The RCA Module initiates the process of assessing how the integrated system is behaving and making any appropriate adjustments to the global FSRM as follows…
Observe	5	Embedded sensor packages monitor the THCS (Pressure, temperature, RH, ...) with some sufficient or optimal sensor placement configuration.
	6	Sensors announce their availability and meta-data.
	7	Sensor data is sent to a static repository in the digital twin via some software interface.
Orient	8	The RCA Module asks the Digital Twin for information about the THCS, its control logic, its components, and available sensor packages.
	9	The RCA Module selects the sensors that it can use to understand the causal relationships between components of the THCS and asks for historical logs.
	10	The Digital Twin interface produces the logs and hands them over to the module.
Decide	11	The RCA Module then analyzes the data using its internal functionality and produces corrected fault-symptom relationships. The module also communicates these corrections to the Digital Twin.
Act	12	The FSRM is appropriately updated in the digital twin, along with metadata about the update (e.g., which agent/version produced it, what uncertainty was communicated, the date/time of the update, etc.).
Observe	13	Sometime later, the Fault Detection Module has detected an anomalous signal in the habitat's THCS, determined that the anomaly is the result of a fault, and triggers the RCA Module to diagnose the root cause.
Orient	14	The RCA Module asks the Digital Twin for information about the THCS, its control logic, its components, and available sensor packages. It also requests access to the FSRM stored on the Digital Twin.
	15	The RCA Module selects the sensors that it can use to determine the cause of the anomalous signal and asks for historical logs.
	16	The Digital Twin interface produces the logs and hands them over to the module.
Decide	17	The RCA Module then analyzes the data using its internal functionality, in concert with the FSRM, to identify a short list of possible root causes along with a probability for each one.
	18	The short list of probable root causes is communicated to the crew via a user interface, to Mission Control via communications link, as well as to the Digital Twin to be stored in its internal historical records.
Act	19	Upon receiving this information, the crew is able to work with MC’s to develop a M&R plan to mitigate the fault.

Cross-System Awareness and Explainability

Motivation and Overview

If the RCA use-case shows how the Digital Twin can be a useful tool for ISHM, this use-case builds off that to highlight how a Digital Twin can assist human users of the system in their understanding of simulation and algorithmic results by providing context and making intersystem connections clear. Digging deeper into the idea of understanding integrated systems, a spacecraft consists of many systems that may be doing very different things but are still highly coupled. A change in any one system may affect how any number of other systems function. In the context of the ISS, the function of a system is monitored by a human controller on the ground who is an expert in that particular system. Even though that controller has had a lot of training in how other systems affect their system of focus, they still may not be experts in adjacent systems. As a result, cross-system awareness may be degraded. A digital twin should be able to make connections between coupled systems possible with increased speed, efficiency, and fidelity, and to communicate these connections to ground controllers, crew members, and autonomous agents onboard the habitat.

Because anomalies may be detected in one system that are caused by a fault in a completely different system, system-of-systems modeling is required to create an effective tool. Holistic system-of-systems modeling for the entire spacecraft may be out of reach for a single computer. As discussed in Chapter 1, the unique mass, power, and transistor density limitations of deep spaceflight restrict the computational resources available for any one computer system. This begins to highlight the need for a hierarchical structure to computation whereby the spacecraft is segmented into tiers of control. A discussion of one way in which this hierarchical structure may manifest follows and is depicted in Figure 5.

At the top of the hierarchical structure is a Vehicle Systems Manager (VSM) which contains a library of software modules necessary for the autonomous functioning of the spacecraft. Among those is a Digital Twin of the spacecraft at large. The Vehicle Digital Twin (VDT) contains the global FSRM that was discussed previously and any global information models. The VDT also contains high-level simulation models that simulate how all the elements of the spacecraft function together which take as input, data from lower-tier Digital Twins by submitting simulation requests and information queries. The VDT must contain sufficient physical and software interfaces to be utilized by humans, robotic agents, and any autonomous software module in the VSM.

The next control tier happens at the element level. Elements are the bigger modules that make up the integrated vehicle and may include a habitat module, laboratory module, or Power and Propulsion Element (PPE). Each element has an Element Systems Manager (ESM). Like the VSM, the ESM would consist of a library of autonomous software modules and a Digital Twin of the element. The Element Digital Twin (EDT) would contain element level information models and simulation models that simulate how all the different systems of that element function together. The EDT fulfills simulation requests and information queries from the VDT and takes as input, data from system level Digital Twins.

Each element is made up of a number of systems that vary depending on the function of that element. Previously discussed systems that are common to human rated spacecraft are ECLSS and EPS but may include many others such as a Fire Protection System, Communication System, or Attitude, Determination, and Control System. Each system may have a dedicated System/Process Manager (SPM) which, like above, contain autonomous software modules and a Digital Twin of the system. The System Digital Twin (SDT) would contain system level information and simulation models. Simulation models at this level should be high fidelity, focused on individual processes, and based on first principles plus empirical statistics. They take as input, direct data from sensor packages optimally placed throughout the system. The SDT fulfills simulation requests and information queries from the EDT. Figure 5 shows a high-level breakdown of the hierarchical computation structure just discussed

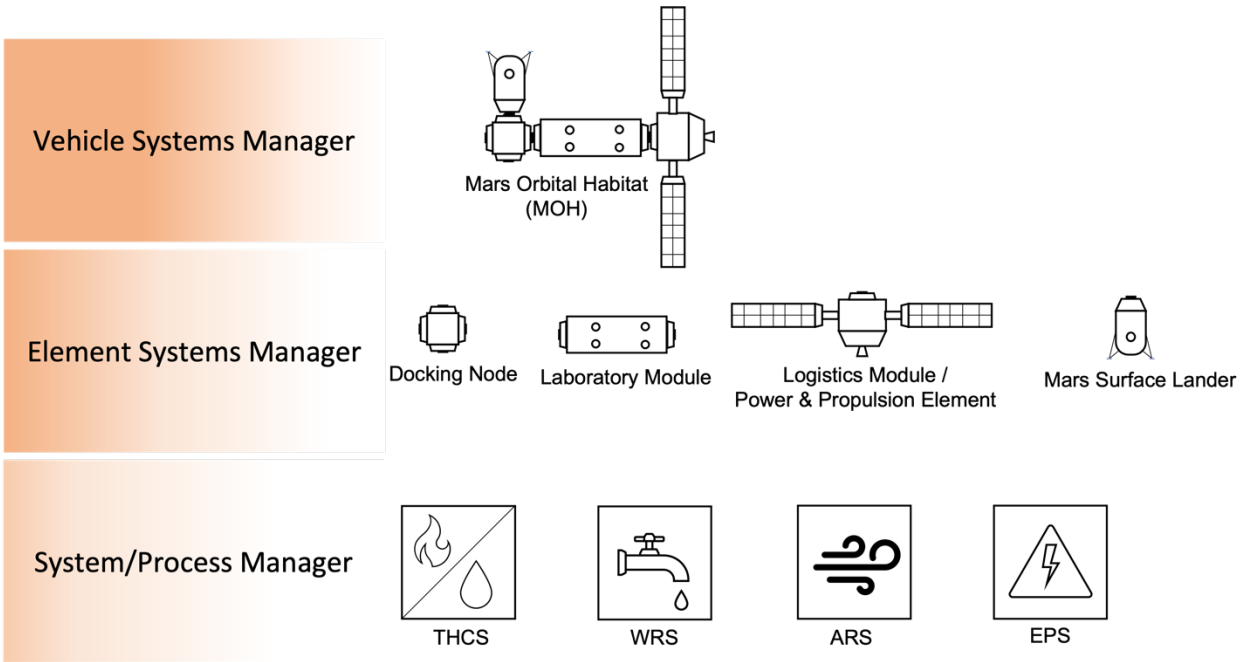

Figure 5: High-level depiction of proposed hierarchical computation structure.

In the following example, an ECLSS MC on Earth is monitoring telemetry from the spacecraft and sees anomalous readings from an O2 sensor. That controller may immediately assume that there is something wrong with that sensor. Let's say that in truth, there is an issue with the EPS which is causing unsteady power to be delivered to that sensor package. The connection will likely eventually be made on the ground, but there will be a time delay due to the nature of communication between humans with different knowledge bases, such as an ECLSS MC and an EPS MC. The hierarchical structure of federated Digital Twins should be able to make connections between interconnected systems quickly and efficiently and communicate to humans in-the-loop the true nature of the detected fault.

Detailed Use-Case Description

Use Case	Cross-System Awareness and Explainability
Subject Area	Fault Management
Primary Actor(s)	Fault Detection Module, RCA Module, State Estimation Module
Precondition 1	There is an Atmosphere Revitalization System onboard the Laboratory Module, a major habitable element of a Mars Orbital Habitat.
Precondition 2	The ARS has an O_2 sensor at the intake for the CO_2 removal sub-system.
Precondition 3	The Laboratory Module also has an Electrical Power System. The EPS has a voltage meter monitoring the output of a Power Distribution Module.
Precondition 4	Both the ARS and EPS have independent System/Process Managers, a suite of software that is responsible for all the autonomous functioning of the system. They also have their respective Digital Twins.
Precondition 5	The Laboratory Module has an Element Systems Manager, a suite of software that is responsible for all the autonomous functioning of the element, along with a dedicated Digital Twin.
Precondition 6	The Mars Orbital Habitat itself has a Vehicle Systems Manager, a suite of software that is responsible for all the autonomous functioning of the vehicle, along with a dedicated Digital Twin.
Precondition 7	There is a mechanism in place for the passing of information and the communication of requests between the various levels of this computational hierarchy.
Triggers	Fault Detection Module runs at some sample rate OR is triggered at the exceedance of some threshold. The remainder of this scenario is triggered at the determination of the existence of a fault.
Basic Flow	
Description	An anomalous signal from an O2 sensor is investigated by a hierarchical structure of autonomous systems management software and associated

		Digital Twins. The resulting RCA, M&R recommendation, and all supporting discovery information is recorded and communicated to MC's on Earth.
Observe	1	An O_2 sensor embedded at the inlet of the ARS monitors oxygen levels entering the CO_2 removal sub-system.
	2	The DAQ supporting this sensor announces its availability and meta-data.
	3	Sensor data is sent to a static repository in the ARS SDT via some software interface.
Orient	4	The Fault Detection Module in the ARS SPM regularly queries and receives up-to-date sensor data from the SDT to monitor for anomalous system behavior.
Decide	5	Via its own internal algorithms, the Fault Detection Module determines that anomalous sensor data coming from the O_2 sensor represents the likely existence of a fault.
Act	6	The ARS SPM reports to the Laboratory ESM that a fault has been detected in the ARS O_2 sensor and provides relevant data to support its findings.
Observe	7	The Laboratory ESM receives the fault report. It also receives a fault report from the EPS SPM that a voltage meter has detected an unsteady voltage at the output of a Power Distribution Module (PDM).
Orient	8	The RCA Module in the Laboratory ESM asks the Laboratory EDT for access to the Laboratory Element FSRM.
Decide	9	The RCA Module then analyzes the data provided by the ARS SPM and EPS SPM using its internal functionality, in concert with the FSRM, and reasons that the most likely cause of the anomalous O_2 sensor data is the unsteady power supplied to the sensor by the faulty PDM in the EPS.
Act	10	The Laboratory ESM sends the RCA results to the ARS EPM along with a time series log of anomalous voltage sensor data and operational parameters from the EPS PDM and requests a confirmation that this could be the cause.
Observe	11	The ARS SPM receives the data and triggers its State Estimation Module to estimate what the O2 sensor output would be given the time series of input voltages supplied by the Laboratory ESM.
Orient	12	State Estimation Module asks the ARS SDT for information about the ARS and its components, along with sensors in it.
	13	State Estimation Module selects the sensors that it can use to estimate the output of the O_2 sensor and asks for historical logs.
	14	The digital twin interface produces the logs and hands them over to the module.
Decide	15	The module then analyzes the data using its internal logic and produces a time series of estimated O_2 sensor outputs given the suspected power supply issue.

16 The ARS SPM interrogates the State Estimation Module output and confirms that the estimated O_2 sensor behavior matches the observed behavior within an acceptable band of uncertainty. A confirmation is sent to the Laboratory ESM.

Act

17 Upon receiving the confirmation from the ARS SPM, the Laboratory ESM sends a notification of the fault, its root cause, and any relevant data sets, metadata, and other information to the MOH VSM.

18 The MOH VSM stores all received data and information about the fault and discovery process in the MOH VDT fault report logs.

19 MOH VSM communicates the results of the investigation to MC's on Earth, including a full explanation and recommendation for a M&R task to fix the faulty PDM.

There are six major software blocks communicating in the above use-case description, along with the physical systems under investigation. For clarity, Figure 6 provides a swim lane diagram of the process. The numbered blocks correspond to the use-case steps outlined above.

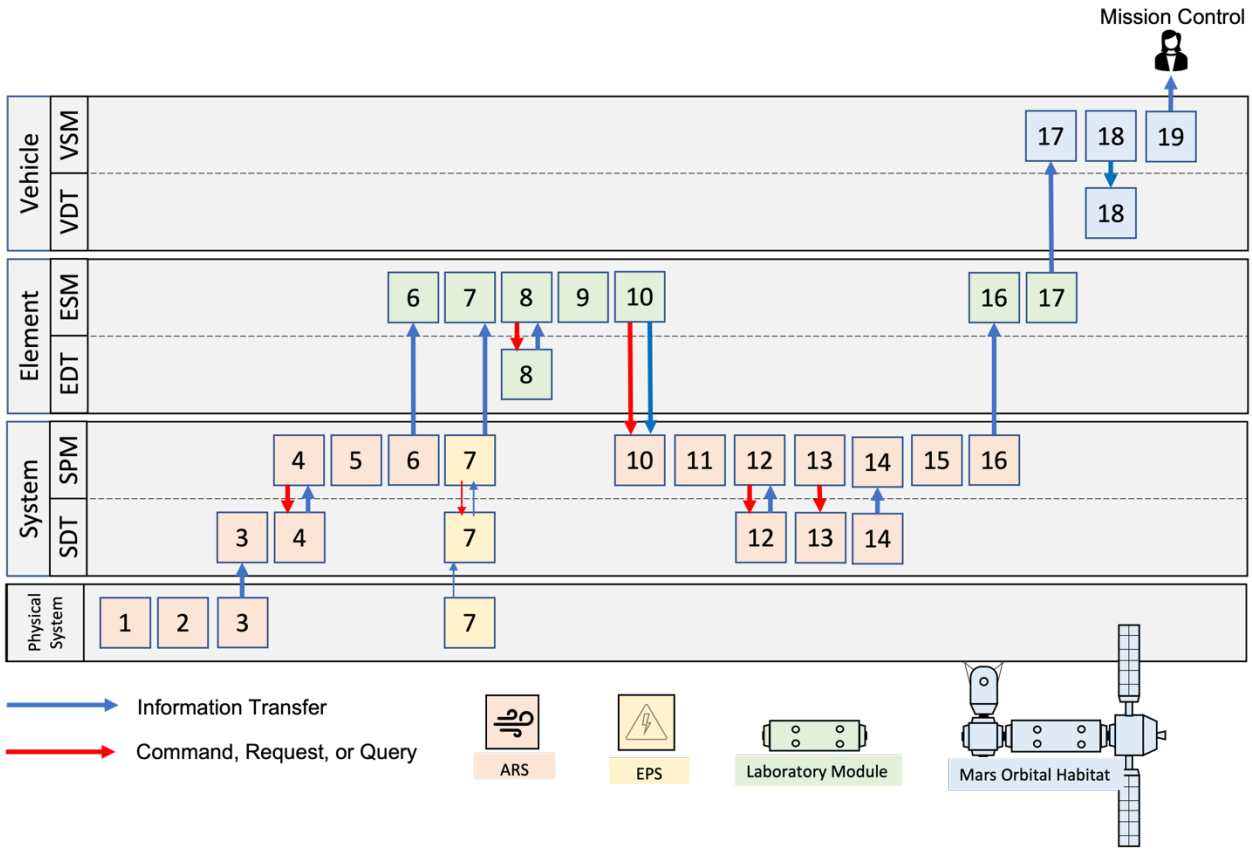

Figure 6: Swim Lane diagram depicting the timeline of the "Cross-System Awareness" use-case. The block colors correspond to the vehicle, element, or system in which the numbered step is occurring.

System Checkout During Prep for Crew Arrival

Motivation and overview

During the un-crewed phases of operation, some systems may be operated at a reduced capacity or powered down completely. As discussed in Chapter 1, ECLSS is an example of a life-critical system that this circumstance may apply to. Any degree of reduced capacity operation may cause components to degrade differently than they would during full nominal operation. Because of this, various system model parameters (i.e. material properties, etc.) and component degradation states may be different upon reboot than they were when the system was shut down. These parameter and state discrepancies would manifest as systemic error in modeling results. The Digital Twin should be able to calibrate itself to match the true state of the system after reboot, as illustrated in Figure 7.

Additionally, due to the nature of spacecraft, systems across a habitat would be highly coupled where a small change in any one system can have an effect in any number of other systems. Therefore, engineering assumptions made to qualify a component, likely fix parameters in other coupled systems. This may be fine during system design, but inappropriate when assessing the capabilities of a safety critical system during dynamic flight operations. Inappropriately fixed parameters can skew simulation results and potentially misinform critical decisions. A Digital Twin will always have the most up-to-date parameter values that reflect the complex dynamics of a spacecraft, particularly one with humans onboard. Normal degradation will also occur for many components during nominal system operation. As the baseline performance changes, modeling errors will accumulate. The Digital Twin should be able to actively adjust model parameters to track dynamic baseline performance of the system.

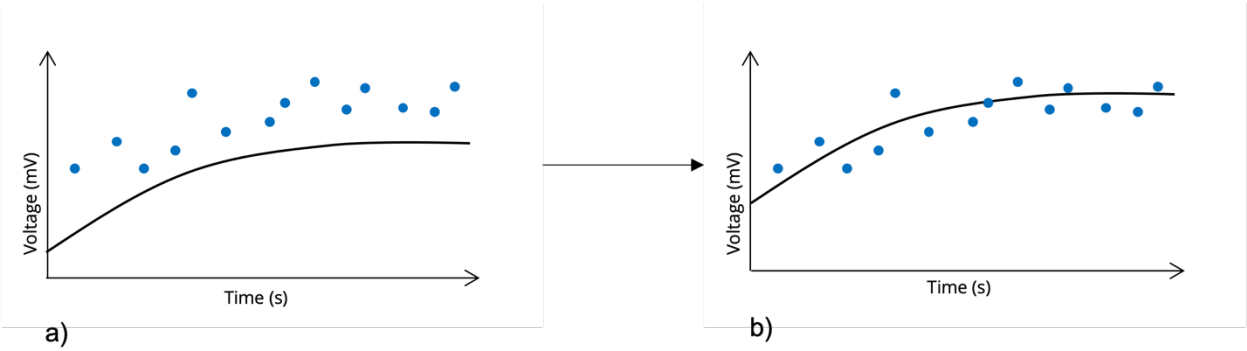

Figure 7: Example of an offset error correction after model calibration. Hypothetical sensor values are depicted by blue dots. State estimate is shown as a black line. a) Estimate based on system state before shutdown has an offset error. b) After system reboot, model calibration corrects system state estimation offset error.

An enabling feature of a Digital Twin will be a Model Conformity Enforcement Driver (MCED). The MCED is an internal mechanism responsible for monitoring the delta between information models and simulation results, keeping watch for systemic errors. If a systemic error is detected, the MCED should be able to optimize simulation model parameters such as to minimize the error. This ensures that simulation results will resemble the true function of the physical asset as closely as possible.

There is the potential for a parameter optimization problem to be ill-posed whereby two sets of parameters result in the same simulation data. The MCED could be made robust to this by first constraining the parameter optimization problems by the physical limits of the components involved and by pre-determined operational rules. Additionally, most equipment under steady-state operation will experience continuous degradation. Meaning that model parameter values will not likely have drifted far from their previous values. Large deviations in system behavior as compared to simulation model behavior will likely be the result of equipment failure or other anomalous condition that should be flagged by fault detection algorithms processed through the ISHM software suite. In the event that the MCED encounters multiple solutions that lie within all applied constraints, it may issue a request for review to human operators in Mission Control.

Confining the authority to update model parameters, without returning a corresponding influence on the Physical Twin, to an internal mechanism prevents those parameters from becoming global variables. Giving this authority to external agents would expose model parameters to the risk of being incorrectly assigned by faulty software. Should an external actor influence the Physical Twin in a way that changes how it should be modeled, it may submit a parameter change through the MCED.

The following use-case will highlight a necessary part of living in a spacecraft, the processing of human urine into usable water. The Urine Processing Assembly (UPA), part of the Water Recovery System (WRS), is responsible for this task. The UPA has a series of peristaltic pumps that transfer pre-treated urine from the Wastewater Storage Tank Assembly to the Distillation Assembly. The peristaltic pump was selected for this application because it can deal with two-phase flow (both compressible and non-compressible constituents) (Williamson, et al. July 2019). This type of pump works by cyclically compressing a polymer tube between a series of rollers, pushing both gasses and fluids through the tube.

Because there is no urine to process when humans are not present, the UPA is a great example of a system that will likely be shut down during periods of dormancy. During an extended period of inoperation, the tubing is left in a statically deformed condition between the rollers,

losing its elastic memory. It is likely that when a new crew does arrive, any peristaltic tubing may be in a degraded state, and pumping capacity potentially reduced from what it was when the system shut down. The parameters used to model this system will likely need to be updated to accurately estimate its performance.

Detailed Use-Case Description

Use Case	System Checkout During Prep for Crew Arrival
Subject Area	System characterization, model calibration, simulation
Primary Actor(s)	
Precondition 1	The habitat is uninhabited but there is a crew about to leave Earth.
Precondition 2	The ECLSS system has been largely powered down or operating in some "safe" mode with reduced capacity.
Precondition 3	The ECLSS system has a Urine Processor Assembly (UPA) containing peristaltic pumps.
Precondition 4	There are sensors monitoring the flow rate through the peristaltic pumps.
Precondition 5	The Digital Twin has an internal Model Conformity Enforcement Driver (MCED) which has access to both information and simulation models.
Precondition 6	The MCED has the authority to change the value of model parameters within the Digital Twin.
Triggers	System reboot from dormancy triggered by a command from Mission Control. Model parameter calibration triggered internally to the Digital Twin by system controller state change.
Basic Flow	
Description	The habitat orbiting Mars has been in a dormant state for a couple years. During this time of dormancy, some systems have been operating at a reduced capacity. A new crew is preparing to leave Earth and in anticipation, Mission Control will send a command to the habitat to ramp up all dormant systems to full operational capacity such that they can be checked out to assure the habitat is safe before the crew launches. 1 Mission Control signals the habitat to ramp up all systems to full operational capacity. 2 The Habitat VSM receives the signal and proceeds to distribute instructions to the appropriate indentured systems management software throughout the habitat. 3 The ECLSS SPM receives instructions to ramp up its subsystems to full operational capacity. 4 The UPA controller is triggered to ramp up and the system begins operating.

	5	A human urine simulant that had been previously stowed for this purpose, is processed through the system. As it does, data from embedded sensor packages are logged in the appropriate repositories in the Digital Twin via some middleware software.
	6	The UPA controller state change from off to on triggers the MCED in the ECLSS Digital Twin to perform a post-dormancy model calibration.
Observe	7	The MCED runs a simulation of the UPA given all of the model parameters that existed before the system shut down.
Orient	8	The MCED compares the simulated sensor data to the real data coming in from the sensor package and calculates an error.
	9	The MCED detects systemic error between the simulated and real pumping capacity of one of the peristaltic pumps.
Decide	10	Through its internal algorithms, the MCED optimizes the model parameters such as to minimize the detected error. In this case that parameter may be the elasticity of the peristaltic pump tubing.
Act	11	The resulting optimized model parameters are then updated in the Digital Twin.
	12	With simulation models now sufficiently calibrated to match the true function of the system, the Digital Twin is ready to be used by external agents for ISHM checkout.

Safety-Critical What-If Questions Without Risk to Hardware

Motivation and Overview

Besides the ability to forecast what will go wrong and when given current operational parameters, perhaps one of the most useful prospects of a Digital Twin would be the ability to know what could go wrong and when, should the operational parameters change. This is the domain of the What-If Engine (WIE). The WIE is not part of the Digital Twin but exists as an external software module in the user space. Like other external software discussed, this work does not prescribe details of its internal mechanisms but does give an overview of its benefit and explores how the WIE would interact with the Digital Twin to perform its functions. The WIE would have two primary functions; 1) to assess the state of systems given planned future operational parameters at the request of a human operator (Mission Control or crew), and 2) to continuously look ahead for potential problems given a locus of feasible operational parameters derived from the habitats operational use envelope and historical usage data.

Keeping humans alive in deep space is no simple task. The crew cannot afford to have problems with safety-critical systems. Therefore, operational decisions are not made lightly. Decision makers would like to know ahead of time what the impact of a decision is going to have on the state of the habitat. Because a Digital Twin enforces simulation models to always have the most up-to-date parameter values that reflect the complex dynamics of a spacecraft, it has the potential to be a powerful tool for mission planners in forecasting system behavior. This is the first function of the WIE and the subject of the use-case description below. A mission planner, that is any agent in a position to make a decision that affects the mission (whether that be an MC, crew member, or other autonomous agent), may engage the WIE to investigate whether any issues will arise as a result of that decision. The WIE can then request simulation instances from the Digital Twin given a corresponding change in operational parameters. The Digital Twin would run the requested simulations and deliver them back to the WIE. The WIE would then use its own internal algorithms to determine any risks that may exist and communicate that to the mission planner.

The second function of the WIE is not triggered by an external actor, but rather runs continuously in the background or at some reasonable sample rate. In this scenario, the WIE is constantly looking ahead and trying to anticipate the most likely ways in which the operation of the habitat may change. It requires that the Digital Twin contain information about the operational envelope of the habitat and its systems as well as a repository of historical use data. Some degree of machine learning will be required of the WIE whereby it can look at this data and learn how mission planners are commanding the habitat and its systems and develop likely future scenarios. For safety-critical systems, the WIE should additionally consider worst-case scenarios that address the boundaries of the operational envelope. It would then need to perform some sort of sensitivity analysis for these scenarios to understand what operational parameters would most likely be affected and by how much. The WIE should be able to utilize the simulation models stored in the Digital Twin to simulate hypothetical scenarios given determined simulation model parameters. The end product of this process would be to alert mission planners to potential issues “should this state of operation occur” and suggest specific operational parameter changes or preventative maintenance tasks that would mitigate any issues. This is a bit of a higher hanging fruit than the first functionality of the WIE but would clearly have a large contribution to the safety and autonomy of the habitat.

The process of developing this space of potential scenarios to be investigated by the WIE is an issue that needs to be addressed. Although algorithmic details are out-of-scope for this work, Downward Counterfactual Reasoning (DCR) may be a good starting place. DCR asks the question, “given a known past event, how could the situation have deteriorated if things happened differently?” (Woo 2019) outlines a case for using an incremental DCR search for

discovery of possible extreme natural hazard events, e.g., volcanic, tsunami, or seismic events. This is to be used as a tool to inform natural hazard risk analysis and preparation. It is clear that there is a potential for Woo's methods to be applied to deep-space hazards and spacecraft failure events. The process could begin by first looking at past instances where the system was in a similar state to the one it is currently in. A DCR search process could be implemented where the WIE would find ways in which events might play out in a manner that is worse than they actually did, all while enforcing causal relationships between events and systems. Adding stochastic variation to current system states before implementing the DCR search may yield even more unexpected events. This process is repeated incrementally with increasing losses from scenario to scenario until possible worst-case or disastrous events are discovered.

The following use-case describes a hypothetical situation where a new crew of four is in transit to the Mars System onboard the Mars Transfer Vehicle (MTV). The four crew members onboard the MTV will be referred to as crew A, B, C, and D. As part of a routine health management regimen, crew A and B are scheduled to exercise today, and crew C and D are scheduled to exercise tomorrow. Unfortunately, a coolant leak has been detected in the Orion Spacecraft's Thermal Control System (TCS) requiring immediate attention. Crew B must attend to repairs today and forego the planned exercise. Crew B must instead exercise with crew C and D tomorrow. Of course, the more people exercising, the more CO_2 is being exhausted into the cabin atmosphere. Before officially re-scheduling Crew B to exercise tomorrow, Mission Control would like to know if the CDRS can safely handle the extra load.

Detailed Use-Case Description

Use Case	Safety-Critical What-If Questions Without Risk to Hardware
Subject Area	System state forecasting, model parameter optimization, simulation
Primary Actor(s)	What-If Engine (WIE)
Precondition 1	The MTV has a VSM, a suite of software that is responsible for all the autonomous functioning of the vehicle. This includes a vehicle-level WIE and interface with external communications system.
Precondition 2	The MTV has a dedicated VDT that has knowledge of the current configuration and operational state of its elements including how they are connected and interact with each other (vehicle level FSRM). It also contains repositories of historical operation data.
Precondition 3	The Habitat Module has an ECLSS containing a fully functional ARS and WRS. It also hosts exercise equipment for the crew.
Precondition 4	The Habitat Module has an ESM, a suite of software that is responsible for all the autonomous functioning of the element, along with a module-level WIE.

Precondition 5		The Habitat Module has a dedicated EDT that has knowledge of the current configuration and operational state of its systems including how they are connected and interact with each other (an element level FSRM).
Precondition 6		Both the ARS and WRS have independent SPMs, a suite of software that is responsible for all the autonomous functioning of the system. The respective SPMs include a system-level WIE.
Precondition 7		The ARS and WRS have dedicated SDTs that have knowledge of the current configuration of their respective systems including the components in them, their attributes, the sensors in them, etc. It also has simulation models with the most up-to-date model parameters reflecting the function of the systems.
Precondition 8		There is a mechanism in place for the passing of information and the communication of requests between the various levels of this computational hierarchy.
Triggers		Runs at the request of a mission planner.
Basic Flow		
Description		Unexpected circumstances have necessitated that three people exercise at a time (hypothetical flight rules allow only two). Mission Control has asked the WIE in the Habitat Module ESM to investigate whether the CDRS can safely handle the extra load.
Observe	1	The MTV VSM receives a command from Mission Control to investigate if the CDRS in the Habitat Module can handle three crew exercising at a time.
Orient	2	The WIE in the MTV VSM queries the MTV VDT for sensor data with the arguments: crew is exercising, and there is a change in the sensor data during that time.
	3	The VDT has operational logs and can come up with a time series for a period of time when crew is exercising. It then uses that time series and looks for sensor data that changes significantly from its baseline. The VDT delivers the applicable sensor data logs to the VSM.
Decide	4	After receiving the logs, the VSM WIE processes the data logs and finds that there is a X% spike in cabin CO_2 levels and Y% spike in cabin humidity per crew member exercising. It also finds that potable water consumption is increased by Z% and an increase in wastewater entering the UPA following the exercise period.
Act	5	The MTV VSM sends a request to the Habitat Module ESM to investigate the CDRS. It also passes along all the information is has received and deduced.
	6	The Habitat Module ESM passes the request and information along to the ARS SPM.
Orient	7	Upon receiving the request and information, the ARS WIE analyzes the information package and through its internal algorithms, develops a series of model parameters appropriate for modeling the CDRS under the intended conditions.

	8	The ARS WIE then sends a request to the ARS SDT to run a simulation of the CDRS given the derived operational parameters.
	9	The ARS SDT runs a simulation with the given parameters and returns the simulation results to the ARS WIE.
Decide	10	The ARS WIE analyzes the simulation results through its internal algorithms and predicts that the CDRS will be capable of processing the added CO_2 and humidity load. It also calculates an uncertainty and margin of safety.
Act	11	The ARS WIE communicates these results including uncertainty and safety margin to the Habitat Module ESM.
Orient	12	While the ARS SPM was processing steps 7 – 10, the WIE in the Habitat Module ESM was using those computation cycles to determine if there are any other systems that might be affected by the intended operational conditions. The ESM WIE requests access to the element level FSRM in the Habitat Module EDT.
	13	The Habitat Module EDT connects the ESM WIE to the FSRM.
Decide	14	Through its internal algorithms, the ESM WIE uses the FSRM to determine that there is a causal relationship between the intended operational conditions and the Habitat Module WRS.
Act	15	The Habitat Module ESM sends request to the WRS SPM to investigate the effect that intended operational conditions would have on the WRS. It also passes along all relevant information and data.
Orient	16	Upon receiving the request and information, the WRS WIE analyzes the information package and through its internal algorithms, develops a series of model parameters appropriate for modeling the WRS under the intended conditions.
	17	The WRS WIE then sends a request to the WRS SDT to run a simulation of the WRS given the derived operational parameters.
	18	The WRS SDT runs a simulation with the given parameters and returns the simulation results to the WRS WIE.
Decide	19	The WRS WIE analyzes the simulation results to predict that the WRS will not be capable of processing the additional wastewater intake from all sources (perspiration, urine, cleaning wastewater, etc.).
	20	The WRS WIE then uses its internal algorithms to determine an optimized set of model parameters that would increase the efficiency of the WRS. In this case, the operating temperature of the WRS Catalytic Oxidation Reactor will need to be increased. The Catalytic Reactor had been operating at a lower temperature to increase the RUL of its O-ring seals.
Orient	21	The WRS WIE then sends a request to the WRS SDT to run a simulation of the WRS given the optimized operational parameters.
	22	The WRS SDT runs a simulation with the given parameters and returns the simulation results to the WRS WIE.

Decide

23 The WRS WIE analyzes the simulation results through its internal algorithms and predicts that the WRS will be capable of processing the additional wastewater intake if the operating temperature of the Catalytic Oxidation Reactor is increased. It also calculates an uncertainty and margin of safety.

Act

24 The WRS WIE communicates these results including uncertainty and safety margin to the Habitat Module ESM.

25 The Habitat Module ESM passes the conclusion of its study and all relevant information along to the MTV VSM.

26 The MTV VSM stores all received data and information as well as information about the entire process in the MTV VDT historical logs.

27 MTV VSM communicates the results of the investigation to MC's on Earth that the CDRS will be able to handle the additional CO_2 load from three crew exercising together. It also communicates a recommendation to pre-emptively increase the operating temperature of the Catalytic Oxidation Reactor in the WRS. A full explanation is provided including uncertainties and margin of safety.

There are six major software blocks communicating in the above use-case description. For clarity, Figure 8 provides a swim lane diagram of the process. The numbered blocks correspond to the use-case steps outlined above.

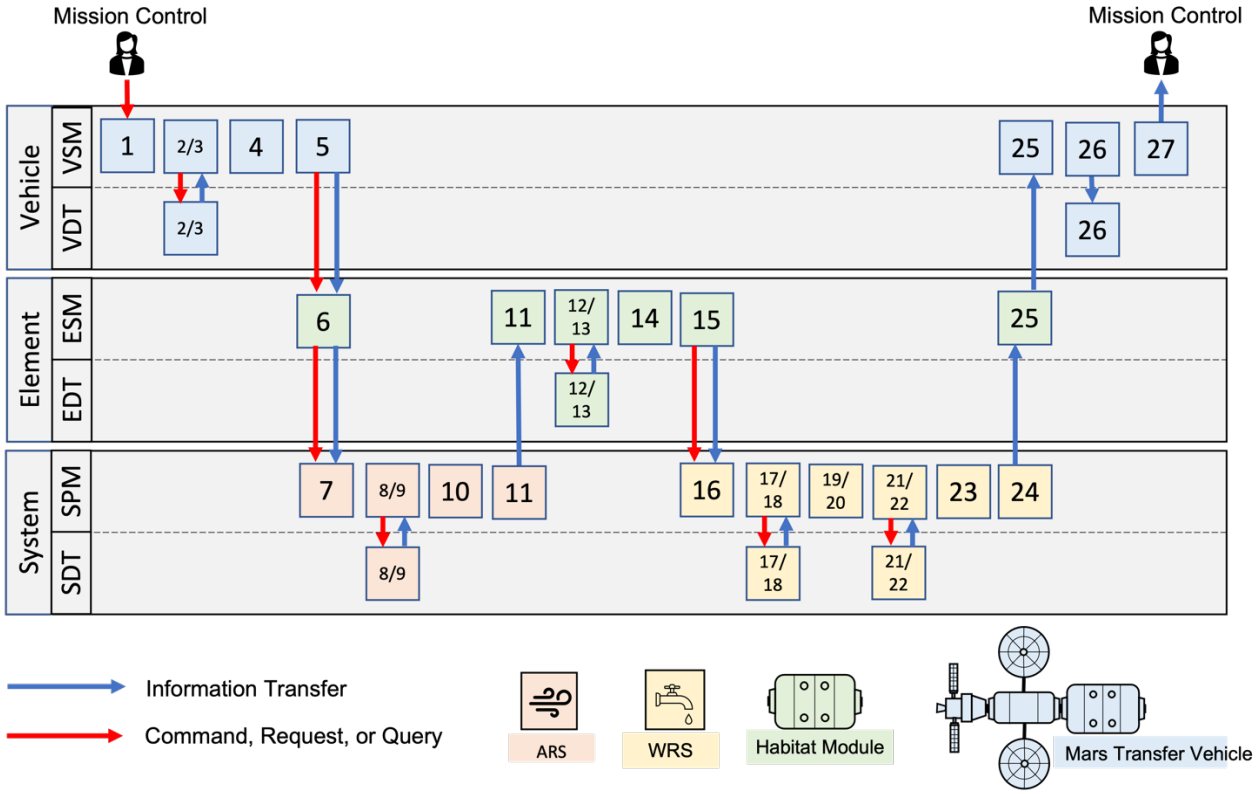

Figure 8: Swim Lane diagram depicting the timeline of the "Safety-Critical What-If questions" use-case. The block colors correspond to the vehicle, element, or system in which the numbered step is occurring.

Global Map Keeping

Motivation and Overview

HOME envisions a deep-space habitat that would be occupied for only about 20% of its operational lifetime (Pischulti, et al. May 11, 2020). Still, despite being unoccupied for the majority of the time, the habitat must be maintained in a safe-for-Humans condition so that when a crew does arrive, it's ready to go. That requirement drives the need for robotic agents that are capable of performing the majority of M&R tasks without the aid of Humans. When a robotic agent is given a task to perform, it must perform that task by means of a series of physical translations that are planned according to its understanding of the accessible environment through which it must move. The environment is made dynamic by the natural imprecision and inconsistency of Humans. Tools and other items are often not put back in exactly the same place. The lack of gravity compounds this issue as any unsecured items are now allowed to free-float about the habitat. An additional source of obstacles are those left behind by other robotic agents. If one agent were to move items out of its way to perform a task, for example, those moved items may end up obstructing the next robotic agent trying to navigate through that space. A Digital Twin can provide a centralized Global Map of the accessible volume of the habitat to warm-start their motion and task planning. The idea is to provide a global map that is accessible, interpretable, and updateable by all robotic agents.

First, it will query the digital twin for the most up-to-date map of the relevant space it intends to pass through. The digital twin must understand what the robotic agent is asking for and then deliver the relevant map data. That is, not a map of the entire habitat, but only the localized part that is necessary. This is to reduce the cost to computational and data storage resources. Also, this map must be interpretable to the querying agent. The robotic agent would then use that map data to plan its movements. As the agent goes about performing its planned motion, it is observing its environment for un-for-seen obstacles. If it should encounter such an obstacle, the agent will replan to avoid the obstacle. That obstacle represents a map discrepancy. The agent must now be able to update the Global Map in the digital twin to correct the discrepancy. The next agent can then "warm start" its task/motion planning with the most up-to-date Global Map.

Detailed Use-Case Description

Use Case	Global Map Keeping
Subject Area	Global map keeping/robotic motion and task planning

Primary Actor(s)		Mobile robotic agent
Precondition 1		There are robotic agents with the means of translating about the accessible internal volume of the habitat.
Precondition 2		There is some task that requires a robotic agent to perform.
Precondition 3		The Digital Twin has a global map of the accessible volume of the habitat already stored in repository.
Precondition 4		The robotic agent has the means of connecting with, and communicating with, the Digital Twin.
Triggers		Some task is required of a robotic agent at a location that is different from the robotic agent's current location.
Basic Flow		
Description		A robotic agent should be able to query for, and receive, the most up-to-date map of the accessible volume of the habitat through which it must translate to a task site. It must also be able to update the global map when a discrepancy is found in the actual environment.
Observe	1	The robotic agent receives or learns information about the location of the task it must complete.
Orient	2	The robotic agent initiates a connection to the Digital Twin via Wi-Fi, Bluetooth, or otherwise.
	3	The robotic agent requests a localized map of the accessible volume of the habitat between its own current location and the location of the task site.
	4	The Digital Twin receives the query and relates the locations translated by the robotic agent onto the global map stored in the Digital Twin.
	5	The Digital Twin partitions a useful portion of the global map and transmits it to the robotic agent.
Decide	6	The robotic agent will use this localized map to warm-start its motion planning to the task site.
Act	7	The agent will then go about performing its planned motion. As it does, it is observing its environment for un-for-seen obstacles. If it should encounter such an obstacle, the agent will replan to avoid the obstacle. That obstacle represents a map discrepancy.
	8	Should a map discrepancy be detected, the robotic agent will transmit information (geometric, location, etc.) about the encountered discrepancy to the Digital Twin.
	9	The Digital Twin receives this information and updates the global map to resolve the discrepancy.

Logistics Tracking

Motivation and Overview

Managing cargo onboard a habitat is a big job. On the ISS, cargo is typically stowed in soft "transfer bags" which are strapped to the walls of various modules. There can be a great number of transfer bags that are usually standardized, meaning they look the same from the outside. During the long periods when the MOH will be un-crewed, robotic agents will need to perform a variety of maintenance and repair tasks in order to maintain the habitat in a safe-for-humans-to-return condition. These M&R tasks will require that the robotic agents be able to access spare parts, tools, and other resources stored in the transfer bags. Exactly which bag contains the supplies needed for a given task is a puzzle that a Digital Twin can assist with.

The digital twin should contain a Global Logistics Manifest of all supplies onboard the habitat with their quantity and locations. The contents of each transfer bag should be logged in a repository in the Digital Twin along with metadata like transfer bag ID, when and on what transit vehicle it arrived, when it was stowed, who stowed it, precise contents, and precise location, etc. The Global Logistics Manifest could be tied to the Global Map to support navigation to the transfer bag location. Transfer bags could also contain a QR code on the exterior of the bag that can be scanned by robotic agents or Humans when they are present. The QR code would have direct access to the Global Logistics Manifest in the Digital Twin and return all metadata associated with that transfer bag.

The Global Logistics Manifest may also be used to inform maintenance task prioritization and planning. For example, one M&R task may be prioritized over another based on spare part availability and the location of task-specific resources relative to the appropriate robotic agent. Once the stowed quantity of a critical resource has fallen below some acceptable threshold, the Digital Twin should be able to automatically trigger a resupply request for that resource.

Detailed Use-Case Description

Use Case	Logistics Tracking
Subject Area	Logistics Tracking, task planning, resource allocation - resupply
Primary Actor(s)	Mobile robotic agent, Maintenance Task Management (MTM) software agent
Precondition 1	Maintenance is due on a piece of equipment that requires a component to be replaced.

Precondition 2	There are spares onboard the habitat stowed in one of many transfer bags.
Precondition 3	Each transfer bag has an identifying QR code visible on its exterior.
Precondition 4	The Digital Twin has a global logistics manifest that includes the contents of all transfer bags already stored in repository.
Precondition 5	There are robotic agents with the means of translating about the accessible internal volume of the habitat.
Precondition 6	There is at least one robotic agent with the means of scanning a QR code identifier as well as opening and manipulating the contents of a transfer bag.
Precondition 7	The robotic agent has the means of connecting with, and communicating with, the Digital Twin.
Triggers	Maintenance is due on a piece of equipment that requires a component to be replaced.

Basic Flow

Description

During MOH dormancy period, most ECLSS systems have been shut down or are operating at some reduced capacity. The Temperature and Humidity Control (THC) system is still operational to mitigate biological growth, corrosion, and electrical issues. An off-nominal pressure differential has been detected across a pump that transports collected water from the THC to the Water Processing Assembly (WPA), triggering a robotic repair response. A robotic agent should be able to query for, and receive, the location of a specific transfer bag containing the required spare part. It must also be able to update the global logistics manifest as to the actual contents of the transfer bag.

1 Through its internal algorithms, a diagnostics software agent concludes that the off-nominal sensor readings are due to a degraded O-ring associated with the pump.

2 Since humans are not present, the diagnostics software agent has triggered a robotic maintenance task to replace the degraded O-ring.

3 A Maintenance Task Management (MTM) software agent sends a query to the Digital Twin requesting the availability of spare O-rings and their location onboard the habitat.

4 The Digital Twin queries its internal Global Logistics Manifest and returns the requested information.

5 The MTM software agent signals a robotic agent to retrieve the spare O-ring from a transfer bag at a location provided by the Digital Twin.

6 The robotic agent translates to the provided location and identifies the target transfer bag.

7 The robotic agent scans a QR code on the exterior of the transfer bag.

8 The robotic agent initiates a connection to the Digital Twin via Wi-Fi, Bluetooth, or otherwise.

9 The robotic agent requests to verify that this is the correct transfer bag and what its contents are.

10 The Digital Twin returns the contents of the transfer bag, verifying that the required spare O-ring is inside.

11 The robotic agent manipulates the transfer bag, opens it, retrieves the O-ring, and returns the bag.

12 The robotic agent indicates to the Digital Twin that an O-ring has been removed from the bag.

13 The Digital Twin updates its Global Logistics Manifest to reflect the contents of the transfer bag.

14 Because the quantity of available spare O-rings is now below a pre-determined safety margin, a resupply request is sent to controllers on Earth.

Chapter 3: Digital Twin Framework

Proposed Framework

Now that some idea of the functional contribution of a Digital Twin to the autonomy and safety of a deep space habitat has been established, let us readdress the Digital Twin Framework abstraction presented in Figure 2. The goal is to define a framework that is sufficiently concise as to guide the tangible instantiation of such a system. There should be enough detail to infer functional engineering requirements which will be addressed later. The section will build up the framework, piece by piece, beginning with the habitat itself.

In the context of this work, the Physical Asset, illustrated in Figure 9 a), is the habitat hardware. Although due to immense complexity and limited in-situ computational resources, it is often necessary to deal with smaller, more easily modeled segmentations of the bigger picture. For that purpose, the habitat can be further broken down into elements, systems, and subsystems, as has been done in the previous chapters. Either way, no matter how the Physical Asset is scoped, it will contain a number of components that will function together via sensors, controllers, actuators, and communication connections to achieve some behavior. That behavior will induce degradation on the components which will in turn affect their baseline function. To monitor this cycle and understand the continuously evolving states of the Physical Asset, there must be some optimal placement of sensors throughout. There must also be a hardware interface(s) through which to interact with the sensors (DAQ) and actuators (controller).

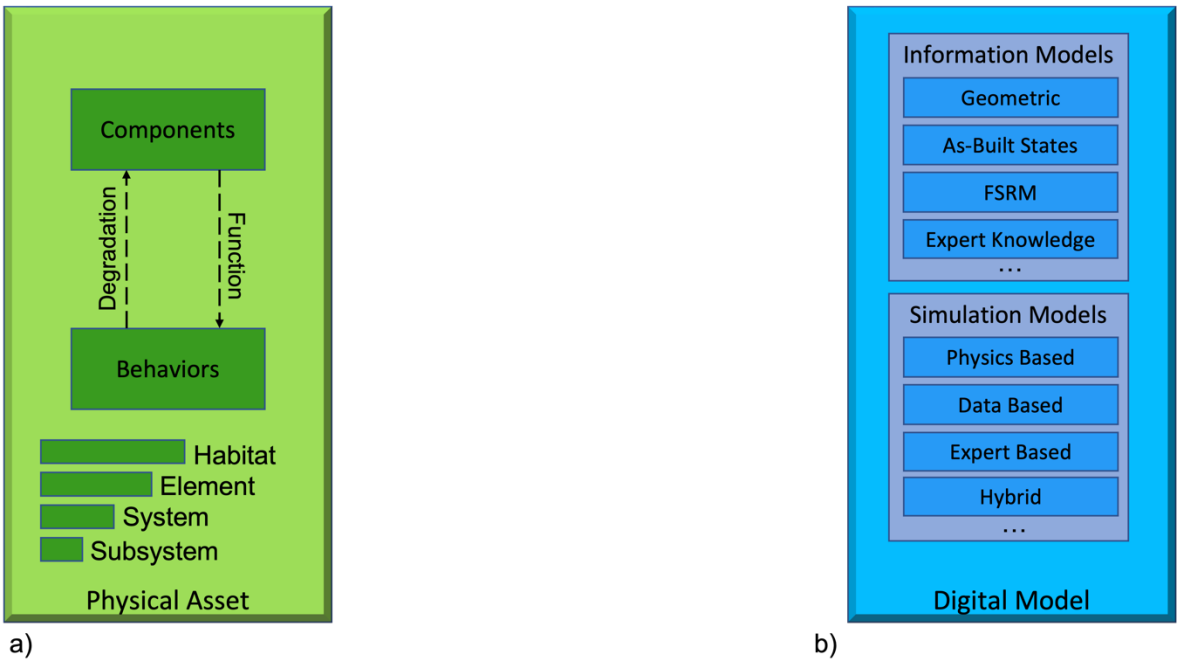

Figure 9: a) Depiction of Physical Asset and relevant details. b) Depiction of Digital Model with relevant details.

We will also need a high-fidelity Digital Model, illustrated in Figure 9 b), of the Physical Asset. This Digital Model will contain various information models. Among others, these information models should include a high-fidelity geometric model, the as-built states of all components within the Physical Asset, a fault-symptom relationship model, and any expert knowledge about the form and function of the Physical Asset. Also required are high-fidelity simulation models. These will be a blend of physics based, data based, expert (empirical), and hybrid models sufficient to fully describe the behavior of the Physical Asset. Since the Digital Model lives in software, there must be a software interface allowing outside agents to interact with it.

As previously established in Figure 1 b), if a one-way data connection is established from the Physical Asset to the Digital Model, allowing for continuous updates of the information model within, the Digital Model is termed a Digital Shadow. Figure 10 illustrates this connection establishing a Digital Shadow which now contains the most up-to-date, current states of the Physical Twin. Because of the complexity of a deep space habitat and consequently, the large number of states being monitored, a Middleware software is established to facilitate the updating of the Digital Shadow. This Middleware can connect to sensor outputs from the Physical Asset's hardware interface and use the sensor data to update the appropriate states within the Digital Shadow via its software interface. The Middleware may also be associated with a graphical user interface, through which Humans can issue updates of the Information model. Human updates may come from the crew or from Mission Control and may encompass updates to the FSRM or expert knowledge models as crew or MC's gain knowledge about the

system or in anticipation of planned reconfigurations. In addition to updating the Digital Shadow, the Middleware also has the function of storing sensor data in an independent Historical Repository. This repository is responsible for logging all the past states of the Physical Asset, leaving space for the Digital Shadow to fully embody its nature as a live, up-to-date representation of the physical system.

As discussed in Chapter 2, there is a need for a Model Conformity Enforcement Driver (MCED). The MCED drives model integration within the Digital Twin by ensuring that all the various models contained therein are reflecting the true nature of the Physical Asset as closely as possible. The conformity of simulation model results with information models and real observations is supported by the MCED's function of monitoring for disparities and updating model parameters appropriately. For example, the MCED may optimize a simulation model parameter to minimize a detected systemic disparity with true sensor values. It was decided that, rather than being an external software agent, the MCED should exist as a mechanism internal to the Digital Twin. This is to safeguard the models therein from being updated un-advantageously by faulty software. Should an external actor influence the Physical Asset in a way that changes how it should be modeled, it may submit a reflecting parameter change through the MCED. Should a parameter change come from a software agent, the MCED should acquire human verification before acting. This would most likely occur in the case of a robotic repair or reconfiguration. It is important to note that the MCED is not a fault detection or diagnosis algorithm. It has no understanding of what constitutes a fault and does not have the agency to directly influence the Physical Asset. It merely enforces conformity among models by comparison and parameter optimization. Fault detection and diagnosis is the realm of external software agents.

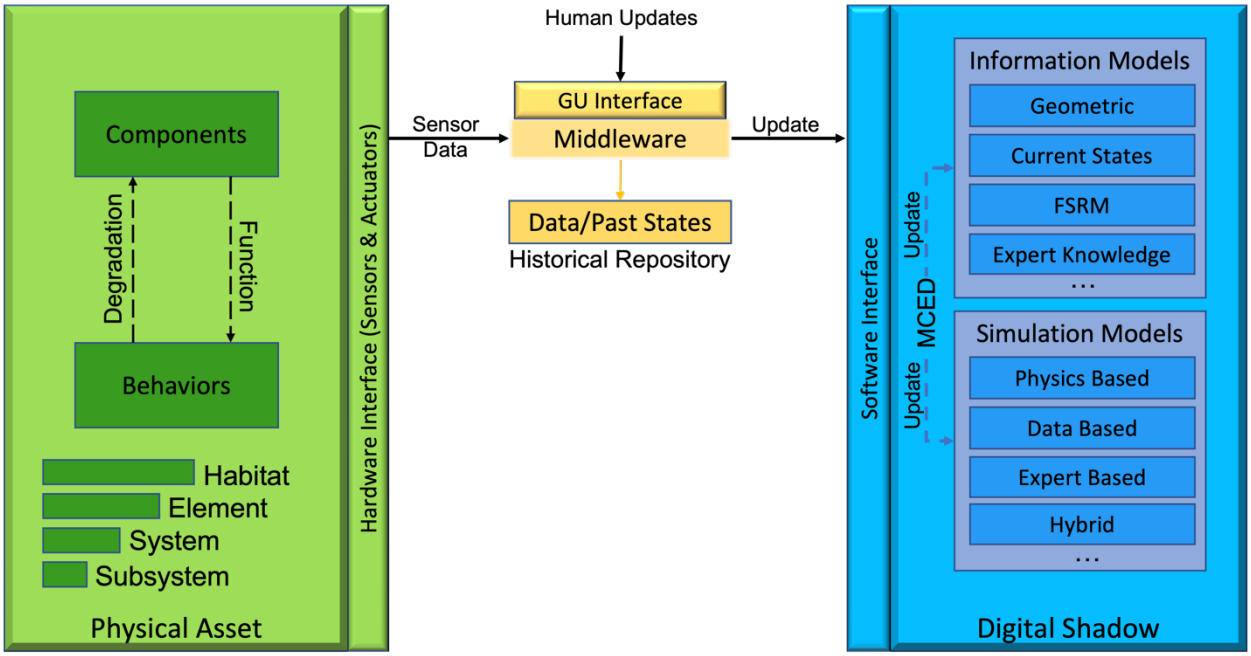

Figure 10: Depiction of a Digital Shadow connected to the Physical Asset through some Middleware software.

As discussed previously, there is little point in understanding the current state of a physical system unless that knowledge is used to benefit the physical system in some way. This is the essence and utility of a Digital Twin. Figure 11 illustrates the establishment of a Return Influence connection via a Supervision Layer, creating the complete Digital Twin Framework. The Supervision Layer encompasses all the agents that would be accessing and utilizing the utility of the Digital Twin. Agents may be Human, robotic, or software-based algorithms. To support vehicle autonomy, it is envisioned that the Supervision Layer agents would exemplify the OODA loop concept previously discussed (Badger, Strawser and Claunch 2019). Each agent is able to access the Digital Twin through its software interface, allowing them to query the Digital Twin for information about the Physical Twin or request simulation results given any set of parameters the agent provides. The Digital Twin must then be capable of processing that request and delivering results in a format that is compatible with the querying agent.

The Supervision Layer has the additional function of logging any analysis or decision results to the Historical Repository along with any pertinent metadata (which agent/version produced the results, what uncertainty was communicated, the date/time of processing, etc.). Traditionally, historical repositories are thought to be integral components of a Digital Twin. However, the thinking within the HOME community places this outside the Digital Twin as a separate entity. This is done primarily due to the limited computational and data transmission resources on a deep-space habitat. Supervision agents may wish to access historical data sets and it is thought

that locating the Historical Repository outside of the Digital Twin would prevent the Digital Twin from becoming a computational bottleneck.

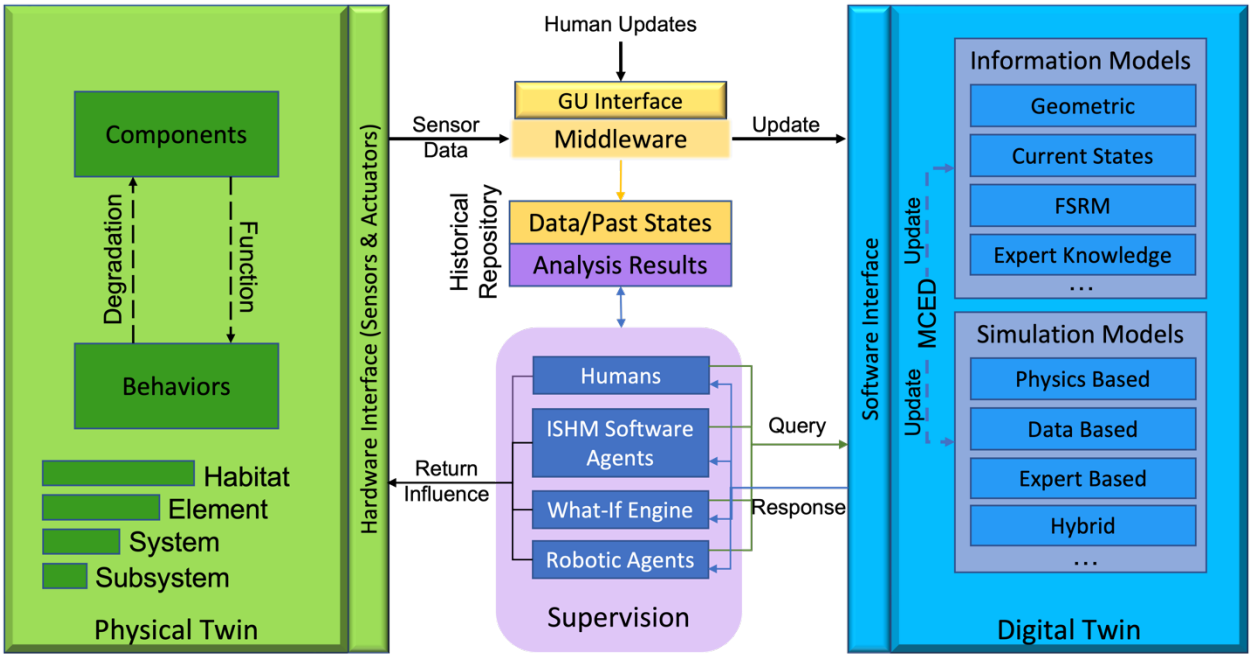

Figure 11: Depiction of the complete Digital Twin Framework. Adapted from *(Gratius, et al. 2023)*.

Functional Requirements

Now that a Digital Twin Framework and functional definition has been established, it is possible to outline concrete functional requirements for such a software. Table 3 and Table 4 present Level 1 and Level 2 requirements respectively for the proposed Digital Twin. Table 5 presents requirements for supporting infrastructure for the proposed Digital Twin Framework.

Level 1 Requirements

Table 3: Level 1 requirements for the proposed Digital Twin.

ID	Type	Text	Rationale
DT-L1-01	Information Model	The Digital Twin shall contain sufficient digital information models to fully describe the current state of the system at any given point in time.	Information models mirror and track the dynamic state of the Physical Asset.
DT-L1-02	Simulation Model	The Digital Twin shall contain sufficient digital simulation models to fully describe the function of the system.	Simulation models are needed to estimate and infer present system states and predict future states given any set of operational parameters.
DT-L1-03	Model Integration	The Digital Twin shall be able to contain and integrate all the various models that describe the system.	Model integration is a primary function of the Digital Twin. It increases the efficiency of simulation processing and trust value of results.
DT-L1-04	External Interface	The Digital Twin shall have sufficient interfaces for the connection and communication with external actors and data sources.	The Digital Twin needs to be able to interface with external software, robotic agents, and humans.
DT-L1-05	Data Exchange	The Digital Twin shall be able to retrieve, package, and deliver requested information from all internal sources to any external recipient.	There is a multitude of information stored within the Digital Twin and an infinite number of potential query requests. It is important to ensure that any querying agent is delivered the data requested and in a format that is understandable and usable by that agent.

Level 2 Requirements

Table 4: Level 2 requirements for the proposed Digital Twin.

ID	Type	Text	Rationale
DT-L2-01	Information Model	The Digital Twin shall contain an updatable high fidelity geometric model of the Physical Asset	A geometric model is used for visualization and context communication for end users.
DT-L2-02	Information Model	The Digital Twin shall maintain a continuously updated log of system states to include all measurable and deducible values of interest as well as commanded operational states of all non-passive subsystems and components.	Ensures the Digital Twin mirrors the true state of the Physical Asset as closely as possible as it evolves dynamically through time.
DT-L2-03	Information Model	System state logs shall be concatenated with useful metadata.	Metadata may include timestamp, data source, operational parameters at time of collection, or any other useful information to define the context of data.
DT-L2-04	Information Model	The Digital Twin shall contain an updatable Fault Symptom Relationship Model (FSRM).	A FSRM is necessary to understand how faults propagate between the systems, sub-systems, and components of the Physical Asset. Particularly useful for Root Cause Analysis and other ISHM algorithms.
DT-L2-05	Information Model	The Digital Twin shall contain a log of expert knowledge to include operational envelopes, decision rules, protocol, or any other human experience derived information about the Physical Asset.	Centrally reposited expert knowledge assists in system health investigations, autonomous decision making, simulation parameter setting, machine learning, and other software logic algorithms.
DT-L2-06	Information Model	Information models shall support multiple levels of abstraction.	To support distributed and hierarchical computation as well as to appropriately scope the response to individual information requests.

ID	Type	Text	Rationale
DT-L2-07	Information Model	The Digital Twin shall be able to track and propagate uncertainties associated with real (sensor) data.	Uncertainty propagation is a necessary part of good data management. This informs better decision making.
DT-L2-08	Simulation Model	The Digital Twin shall contain sufficient physics, data, and expert-based models to provide a high-fidelity description of the nature and behavior of the Physical Asset.	Needed for simulation to predict system states, present and future. To be able to forecast how a system will behave and degrade is the bread and butter of Digital Twin capability.
DT-L2-09	Simulation Model	The Digital Twin shall provide for simulations to be run given any set of parameter values submitted by external agents.	Allows external agents to request specific simulations based on model parameters those agents have deemed interesting without giving permission to permanently after parameters.
DT-L2-10	Simulation Model	There shall be a mechanism for external software agents to submit permanent simulation model parameter changes. Only software agents with appropriate permissions may do this and the Digital Twin shall require human verification before acting.	This may occur in the case of robotic repair tasks or reconfigurations. For example, updating material properties of a robotically replaced O-ring.
DT-L2-11	Simulation Model	Simulation models shall support multiple levels of abstraction.	To support distributed and hierarchical computation as well as to appropriately scope the response to individual simulation requests.
DT-L2-12	Simulation Model	The Digital Twin shall be able to track and propagate uncertainties associated with simulation data.	Uncertainty propagation is necessary for understanding the accuracy of simulation results. This informs better decision making.
DT-L2-13	Model Integration	The Digital Twin shall have an internal mechanism for ensuring that parameter values are consistent across all models.	The consistency of model parameter values needs to be enforced to assure that all models are working from the same baseline of truth.

ID	Type	Text	Rationale
DT-L2-14	Model Integration	The Digital Twin shall have an internal mechanism to monitor for systemic deviations between simulation model results and information model values.	The occurrence of systemic simulation errors likely means that one or more simulation model parameters do not reflect the true nature of the Physical Asset and should be calibrated.
DT-L2-15	Model Integration	The Digital Twin shall have an internal mechanism for calibrating simulation model parameters such as to minimize any systemic error between simulation model results and information model values.	The calibration of simulation model parameters is important to assure that simulation results reflect the true nature of the Physical Asset as closely as possible.
DT-L2-16	External Interface	A software interface shall provide the means for external software agents to discover and connect with the Digital Twin.	External software modules are an intended user of the Digital Twin. These may also include robotic agents.
DT-L2-17	External Interface	There shall be a graphic user interface through which human actors can connect to and interact with the Digital Twin.	Humans are an intended user of the Digital Twin. Crew or Mission Controllers may wish to issue queries or updates directly. The displayed response should be appropriate for a human and easily interpretable.
DT-L2-18	Data Exchange	There shall be a mechanism for external software agents to submit information queries or simulation requests to the Digital Twin and receive responses.	External software agents are intended users of the Digital Twin. They may include software modules or robotic agents.
DT-L2-19	Data Exchange	There shall be a sufficient foundational ontology in place that allows for expected users to be able to communicate clearly with the Digital Twin.	The Digital Twin will have a broad range of users with diverse communication needs. A robust ontology is needed whereby the Digital Twin will be able to interpret the meaning behind any data or queries communicated to it.

ID	Type	Text	Rationale
DT-L2-20	Data Exchange	There shall be an internal mechanism for packaging the human-queried data in a format that is accessible and understandable to the human.	Data formatting for a software user may not be appropriate for a Human.
DT-L2-21	Data Exchange	There shall be a mechanism for external agents to submit content updates to information models within the Digital Twin.	Updates to information model content may be the result of gained knowledge about the form or function of the Physical Asset, real-time sensor data, or from reconfigurations.
DT-L2-22	Data Exchange	There shall be a mechanism for Mission Controllers to update Digital Twin software, wholistically or piecewise.	Software updates may be periodically necessary throughout the lifetime of the Physical Asset. MCs should be able to upload and command updates remotely.
DT-L2-23	Data Exchange	Appropriate permissions shall be enforced for any updates of models, software, or any other aspect of the Digital Twin.	To protect the integrity of the Digital Twin, not all of its aspects should be updatable by all external agents with access to it. External agents should only be able to update aspects of the Digital Twin that are relevant to their function and to which they have been assigned permission.

Supporting Requirements

Table 5: Requirements for supporting infrastructure for the proposed Digital Twin Framework.

ID	Type	Text	Rationale
SUP-01	Physical Asset	The Physical Asset shall be outfitted with sensor packages to measure meaningful state values of sufficient number and placement to accommodate the needs of each intended user of the Digital Twin.	Sensors are the primary source of real data and knowledge about the true state of the Physical Asset and all its parts.
SUP-02	Physical Asset	The Physical Asset shall have sufficient physical interfaces (DAQ or otherwise) to connect all sensors and measurement devices outfitted throughout the Physical Asset to an intermediary software package.	An intermediary software package (Middleware) must be able to access and retrieve sensor data before delivering it to the Digital Twin or Historical Repository.
SUP-03	Historical Repository	There shall be a Historical Repository that acts as an archive for all sensor data, past states, analysis results, and any other information produced by the Digital Twin or its users.	Past states and historical data sets may be required for further analysis by external actors. Archiving this information in a Historical Repository external to the Digital Twin maintains the Digital Twin as a live, up-to-date representation of the physical system while preventing it from becoming a computational bottleneck.
SUP-04	Historical Repository	The Historical Repository shall have sufficient interface to allow external software to query and update as required.	External software agents are the primary users of the Historical Repository. They will need to be able to concatenate relevant databases as more data is produced as well as query historical data sets.
SUP-05	Middleware	There shall be a Middleware software that has access to data sources in the Physical Twin through its interfaces.	The Middleware serves as a conduit for data produced by sensor packages in the Physical Twin and the appropriate databases in the Historical Repository and Digital Twin.

ID	Type	Text	Rationale
SUP-06	Middleware	The Middleware software shall have access to the software interfaces of the Historical Repository and the Digital Twin.	The Middleware serves as a conduit for data produced by sensor packages in the Physical Twin and the appropriate databases in the Historical Repository and Digital Twin.
SUP-07	Middleware	The Middleware software shall have the authority to update Current State information models within the Digital Twin	The Middleware serves as a conduit for data produced by sensor packages in the Physical Twin and the appropriate databases in the Historical Repository and Digital Twin.

NASA Flight Software Requirements

NASA flight software must adhere to additional software engineering requirements outlined in NASA NPR 7150.2D. The NASA Online Directives Information System (NODIS) Library should be consulted for the most up-to-date version of this document. There may be additional requirements that are specific to a NASA organization, program, or other Federal Government agencies. Any implementation of the proposed Digital Twin for a human-crewed spacecraft should adhere to all applicable engineering requirements.

A flight-ready manifestation of the proposed Digital Twin would be classified as Class A: Human-rated space software system. It would also be designated as Safety Critical as defined in NASA-STD-8739.8 as it “Detects, reports, and takes corrective action, if the system reaches a potentially hazardous state.” NASA-STD-8739.8 also outlines Independent Verification & Validation (IV&V) requirements that should be worked into the software development plan. As with any good space mission engineering project, IV&V should be kept in mind from the outset.

Chapter 4: Demonstration Implementation

This chapter will present a simple demonstration of the proposed Digital Twin concept based on the first use-case discussed in Chapter 2, System State Estimation. The demonstration largely follows the work presented in (Torralba, et al. 2022) but with some modification that reflects the improvement of knowledge as the natural course of continued research. Chapter 2 presents the System State Estimation use-case in the context of the CDRS sorbent bed. This context is relevant because a sorbent bed cannot be internally monitored without affecting its performance. State estimation techniques can serve as a tool to create a sort of virtual sensor to gain insight into the state of this unobservable yet safety-critical component.

Physical Twin

The Physical Twin In this study is the Simulation Testbed for Exploration Vehicle ECLSS (STEVE), a physical testbed operated at the University of Colorado Boulder seen in Figure 12. The STEVE testbed was conceived as an analogue for the type of CDRS found on ISS. It consists of a single sorbent bed packed with a 13X zeolite (Kurt J Lasker 13X zeolite) pellets. Zeolite is a microporous mineral. The sorbent bed has two functions: adsorption and desorption of CO_2. During adsorption, CO_2 laden air passes over the pellets. The zeolite's porosity offers a large surface area on which CO_2 molecules adhere. The sorbent bed removes CO_2 from the airflow and the scrubbed air is then returned to the cabin as clean, breathable air. When the zeolite approaches its CO_2 storage capacity, the system switches to desorption. During desorption, the sorbent bed releases CO_2 under thermal vacuum. The CO_2 is directed to other components of the ARS for further processing or vented into space. In the case of the STEVE testbed, the CO_2 is vented to atmosphere.

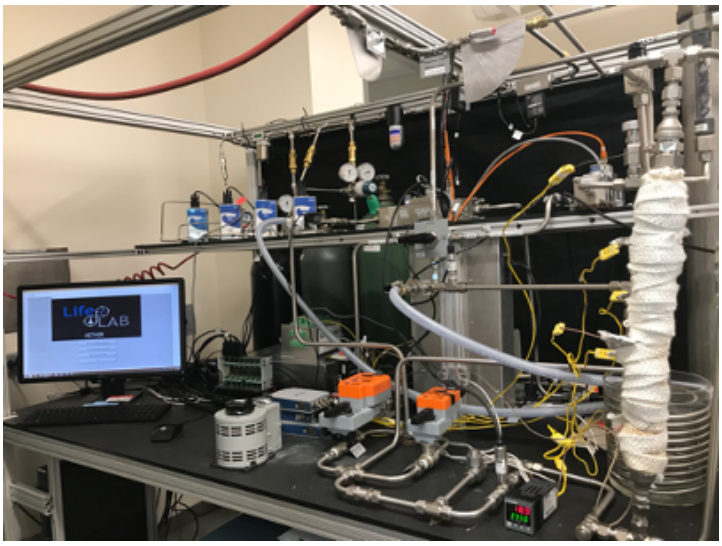

Figure 12: Photograph of the STEVE testbed at CU Boulder used as the Physical Twin in this study. The vertical cylinder on the right of this image is the sorbent bed. It is seen here wrapped in a white insulation. Image Credit: *(Eshima and Purifoy-Frie 2021)*

The STEVE testbed is outfitted with a number of sensor packages that monitor pressure, $\%CO_2$, $\%O_2$, dewpoint, temperature, and relative humidity in both the inlet flow and outlet flow of the sorbent bed (Eshima and Purifoy-Frie 2021). Figure 13 shows a Piping and Instrumentation Diagram of the STEVE testbed showing the locations of components and sensors. The sorbent bed of interest to this study is the red box labeled “13X zeolite bed & heater”.

The STEVE testbed provides a gas mixture, via compressed gas bottles, of nitrogen, oxygen, and carbon dioxide at adjustable percentages. A desiccant bed packed with Drierite beads provides a dew point nominally kept at <= -50℃. A rope heater raises the insulated bed temperature to 200°C and a vacuum pump reduces pressure to below 20 mm Hg for CO2 desorption and regeneration of the pellets via thermal-pressure swing. An automated LabVIEW system commands the flow of gas, valve positions, heater, and vacuum pump from their setpoints for CO_2 adsorption to those needed for desorption. The adsorption/desorption cycle can be repeated for a specified number of cycles per test. The physical interface consists of a LabView/National Instruments DAQ/IO 16-bit system that collects data from the sensors at 2 Hz. LabView publishes collected data in .csv files at a pre-determined rate.

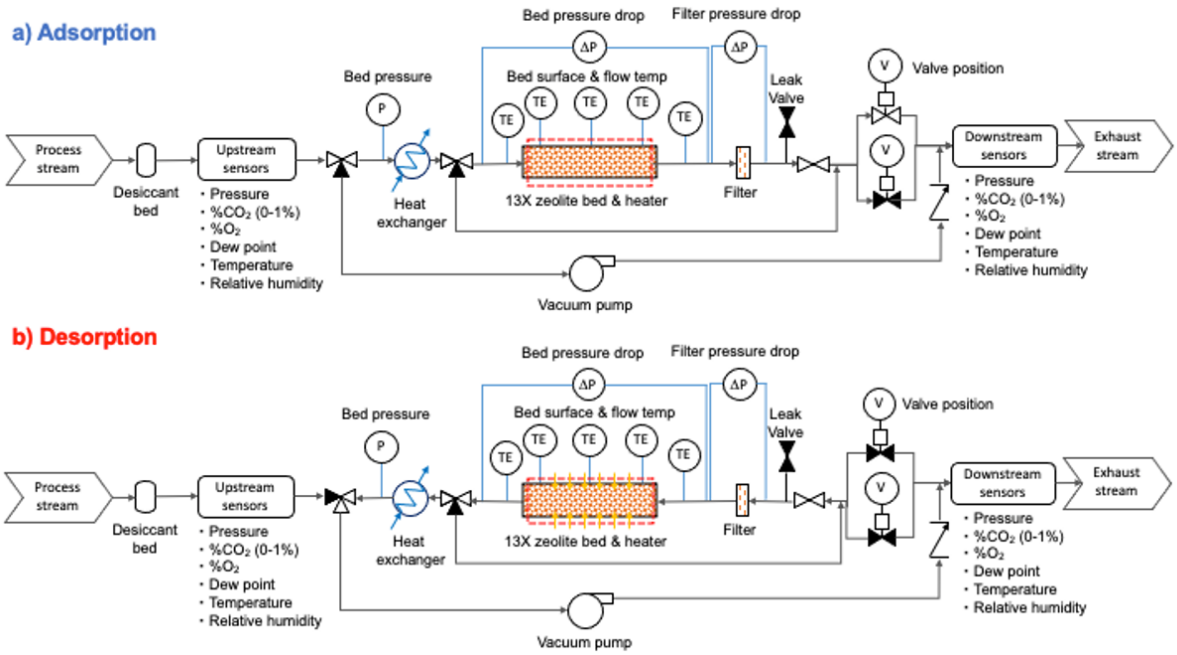

Figure 13: Piping and Instrumentation Diagram of the STEVE testbed. Showing component and sensor locations. The sorbent bed of interest to this study is the red box labeled "13X zeolite bed & heater". Image Credit: *(Eshima and Purifoy-Frie 2021)*

Digital Thread/Middleware

A Middleware software was developed in Python that serves as the Digital Thread connection between the Physical Twin's interface and the Digital Twin. It consists of a script which monitors for the publication of new .csv files by the Physical Twin's LabView software. When a new .csv file is detected, the Middleware script populates a .txt file with this data. The data is structured in three space-delineated columns: timestamp, the data itself, and any communicated uncertainty. A header is provided containing important metadata such as component name, component ID, sensor/state, sensor/state ID, data units, data source, and date of data generation. Figure 14 shows an example of a Digital Twin update .txt file that would be produced. The Middleware software then calls the DTUpdate() function within the Digital Twin, providing the name and location of the .txt file as arguments. It also calls a similar HistUpdate() function within the Historical Repository.

```
SorbentBedInlet
001
Sensor
CO2
%
Testbed
01/03/2021

18000 1.79568652951592 0.001
18000.5 1.79464260205298 0.001
18001 1.79764410961301 0.001
18001.5 1.79922973645454 0.001
18002 1.79696843714251 0.001
18002.5 1.79888972861758 0.001
18003 1.80030254321325 0.001
18003.5 1.80037074839357 0.001
18004 1.80104313251955 0.001
18004.5 1.80014815502873 0.001
```

Figure 14: Example of a Digital Twin update .txt file. From top to bottom, the header lines are: component name, component ID, sensor/state, Parameter, data units, data source, and date of data generation.

The use of .txt files for data exchange was a simple solution for this demonstration looking at a single component but is clearly inefficient and not appropriate for a more sophisticated construct. In practice, a very large number of components, each with their multiple indentured states, would be actively monitored. (Vering, et al. 2019) advocates for the use of MTTQ protocol for such an application. MTTQ is easily scalable for the integration of an enormous number of sensors, actuators, and other end devices.

Digital Twin

A simple proof-of-concept version of the proposed Digital Twin was developed in Python. The main core of the Digital Twin consists of a dictionary of Pandas dataframes, a function for querying the Digital Twin, DTQuery(), and a function that updates the Digital Twin, DTUpdate(). All interaction between the Digital Twin and external actors occurs via .txt file. The DTUpdate() function takes the name and location of a .txt file as arguments. After it locates and opens the .txt file, all information needed to perform the update is contained in the file. The update file contains two primary components: the header, and the data itself. The header tells the DTUpdate() function which dataframe to update as well as some important metadata. The data itself contains three columns: time stamp, data value, and any uncertainty communicated. The DTUpdate() function then updates the appropriate dataframe in the Digital Twin. Figure 15 shows an output of how the information contained in the update .txt file is subsequently stored in the Digital Twin.

	timeStamp	data	uncertainty	compName	compID	sensorState	Parameter	dataUnits	dataSource	dataGenTime
0	18000	1.79568652951592	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021
1	18000.5	1.79464260205298	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021
2	18001	1.79764410961301	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021
3	18001.5	1.79922973645454	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021
4	18002	1.79696843714251	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021
5	18002.5	1.79888972861758	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021
6	18003	1.80030254321325	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021
7	18003.5	1.80037074839357	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021
8	18004	1.80104313251955	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021
9	18004.5	1.80014815502873	0.001	SorbentBedInlet	001	Sensor	CO2	%	Testbed	01/03/2021

Figure 15: Printout of resulting dataframe after update of the Digital Twin with sensor data. Each data point is tagged with the appropriate metadata. This data represents the $\%CO_2$ at the sorbent bed inlet.

The dataframe shown in Figure 15 always contains the most current five seconds of state information. This is a function of the .csv publication rate of the LabView software. Every five seconds, this dataframe is refreshed, ensuring the Digital Twin's integrity as the most up-to-date reflection of the state of the Physical Twin. As discussed in Chapter 3, The Historical Repository serves as an archive of all past states. It is passed an update .txt file in a similar way as the Digital Twin. The difference is that the respective dataframe within the Historical Repository is concatenated every five seconds rather than refreshed. In this way, all past updates are retained for future reference by any external agent.

In addition to DTUpdate(), external agents may also call DTQuery(). The DTQuery() function allows agents to request data and configuration information. It takes the following arguments directly:

DTQuery(0 for data or 1 for configuration, Component ID, Sensor or State ID, Timestamp Start, Timestamp End)

The Sensor or State ID, Timestamp Start, and Timestamp End arguments are only used if requesting data. If data is requested, the returned .txt file would resemble that in Figure 14, providing relevant metadata along with timestamp, data, and any recorded uncertainty values. Figure 16 shows an example of a returned .txt file if configuration is requested. In this file, the querying agent would be able to see which datasets are available. The component name is at the top followed by two lists. The first list shows all sensors associated with the component that are available for access. The second list indicates which states are accessible. For brevity, only $\%CO_2$ concentration is shown as available states. In reality, this list could be quite long depending on how many parameters are being tracked for any given component.

```
SorbentBed

18
Pressure_In
CO2_In
O2_In
DewPoint_In
Temp_In
RelHum_In
Temp_BedFlowIn
Temp_BedSurf1
Temp_BedSurf2
Temp_BedSurf3
Temp_BedFlowOut
Pressure_Out
CO2_Out
O2_Out
DewPoint_Out
Temp_Out
RelHum_Out

5
CO2_SorbentBedInlet
CO2_SBInternalNode1
CO2_SBInternalNode2
CO2_SBInternalNode3
CO2_SorbentBedOutlet
```

Figure 16: Example of what a response to a configuration request might look like. The component name is at the top. The first list shows all sensors available for access. The second list indicates which states are accessible.

The prototype Digital Twin also contains a physics model of the sorbent bed that describes the $\%CO_2$ and temperature profile along the length of the bed. Derivation of this model is beyond the scope of this work. If the reader is interested in exploring the model used here, they can reference (Torralba, et al. 2022) for a full treatment. The model takes parameters like sorbent bead radius and porosity, bed length and diameter, inlet gas composition, inlet gas flow velocity, pressure, and temperature, and cycle time. The model uses values for these parameters stored in the Digital Twin that are optimized to align the model as closely as possible to the true performance of the sorbent bed. Of course, there may be times when a querying agent will request model results based on different parameter values of interest to that agent.

A DTModel() function was written for the State Estimation Module to request model results from the Digital Twin. It takes the distance from the sorbent bed inlet to the node in question as well as the $\%CO_2$ at that location from the previous time step as arguments. Values for any assignable model parameter can be included in the function call as well. Any parameter that is not assigned in the function call will have the value assigned to it from the Digital Twin.

State Estimation Module

The State Estimation Module is a python script developed to estimate the CO_2 mole fraction in the gas phase at discretized nodes along the length of the sorbent bed. Figure 17 shows a diagram of this discretization. Estimating component states at the node points acts as virtual sensors placed at those locations. As previously discussed, this is sometimes necessary when instrumentation is not possible. In the case of the sorbent bed, there are currently no techniques for directly measuring the state of the interior of the sorbent bed without affecting its performance.

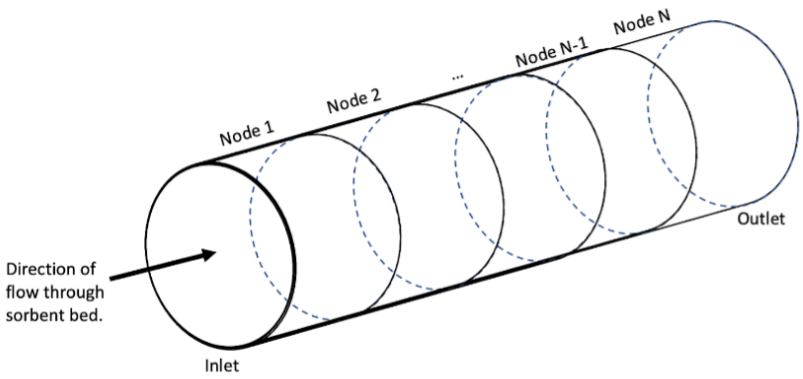

Figure 17: Discretization along the length of the sorbent bed. The Estimation Module estimates the states at each node point, constitute virtual sensors placed at these locations.

Among the options of estimation algorithms that can be applied to this use case, the Kalman filter was determined to be a good fit for this system. Other filter and estimation algorithms are good fits for a more complex suite of ECLSS subsystems such as the Particle Filter. Here, we employed the Extended Kalman Filter due to the nonlinear nature of the cyclic operation of the STEVE testbed and due to the complex system of equations that represent the sorbent bed during adsorption and desorption cycling. A full treatment of the Extended Kalman Filter used can be located at (Torralba, et al. 2022).

The Kalman filter requires system configuration, measured states at the inlet and outlet of the sorbent bed, and model outputs as inputs from the Digital Twin. The State Estimation Module obtains these inputs by calling the DTQuery() and DTModel() functions with the appropriately assigned arguments. After estimation, the State Estimation Module calls the DTUpdate() function to update the estimated states in the Digital Twin as previously demonstrated.

Process

Figure 18 describes how the proposed Digital Twin (shown in green) might interact with external agents (shown in blue). Updates to the Digital Twin or Historical Repository are shown as red arrows. Information that is passed from the Digital Twin to a requesting agent is shown as a blue arrow. Direct communications between external agents are shown as black arrows. The figure is meant to be a high-level depiction of how the Digital Twin framework could be used as part of an Integrated System Health Management package. It is not, however, all-inclusive as any number of external agents could exist here. The full breadth of possible components of a Digital Twin are also not all pictured as data generated from any external agent could be reposited there. The Digital Twin not only acts as a dynamic data repository but may also hold static information such as a high-fidelity geometric model, a Probabilistic Graphical Model (PGM) of system behavior, human-set operational boundaries, and rules and protocol for commanding system parameter changes. The locus of components of a Digital Twin is generally limitless. It is scoped by the needs of its intended users and use-cases.

Figure 19 is a scoped version of Figure 18 specific to this demonstration. The figure shows a step-by-step diagram of the demonstration process. The numbered circles correspond to the steps in Table 6. These steps are based on those which were first developed in the System State Estimation Detailed Use-Case Description in Chapter 2 but modified to reflect this specific demonstration scenario.

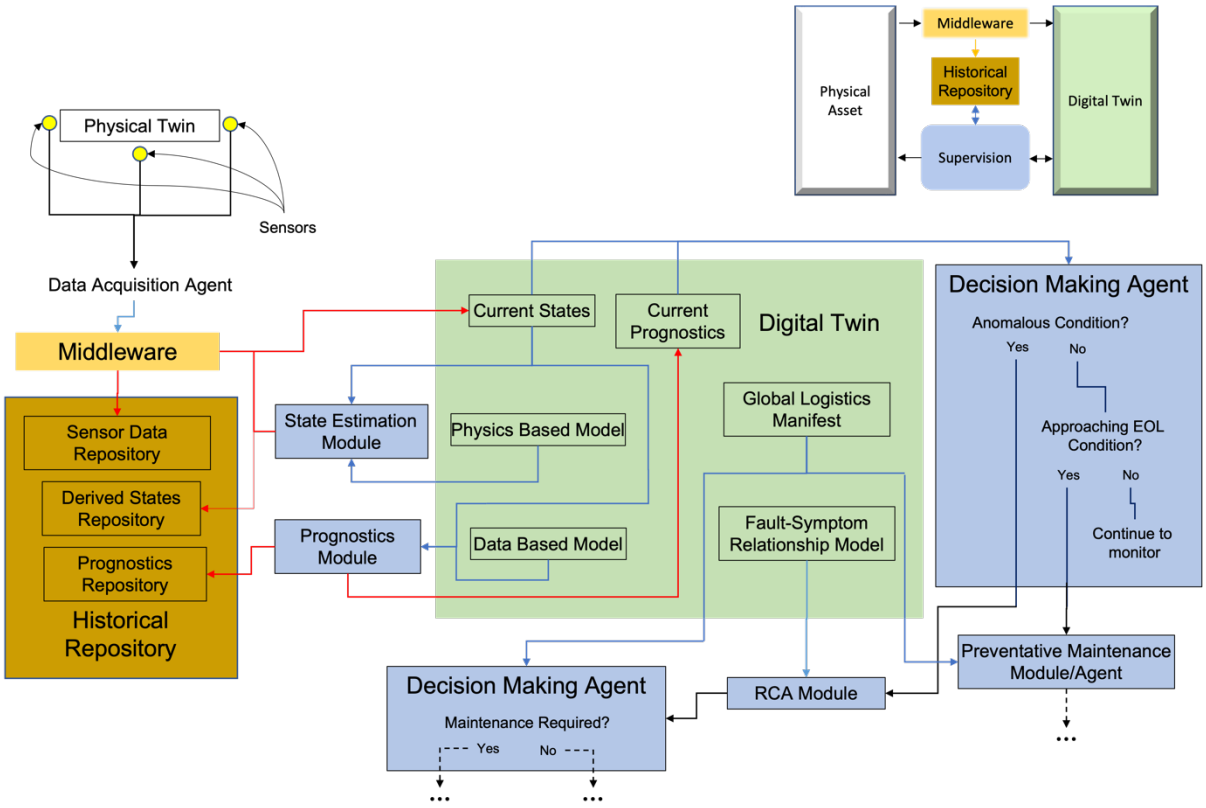

Figure 18: Example of how the proposed Digital Twin (shown in green) might interact with external agents (shown in blue). Updates to the Digital Twin or Historical Repository are shown as red arrows. Information that is passed from the Digital Twin to a requesting agent is shown as a blue arrow. Direct communications between external agents are shown as black arrows. In the top right corner is a color-coded depiction of the proposed Digital Twin framework that maps the elements of this figure to *Figure 11*.

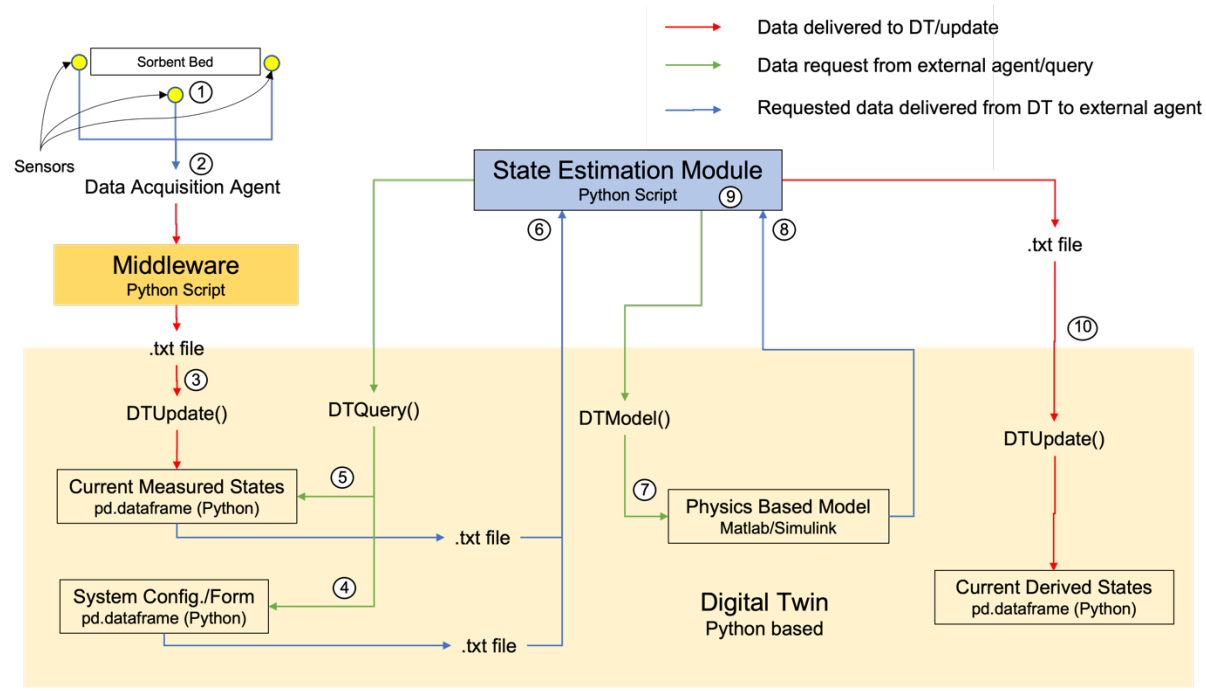

Figure 19: Step-by-step diagram of the demonstration process. All numbered steps correlate to the steps in *Table 6*.

Table 6: Demonstration steps based on those first proposed in the System State Estimation Detailed Use-Case Description.

1	Embedded sensor package monitors sorbent bed (Pressure, temperature, RH, ...) with some sufficient or optimal sensor placement configuration.
2	Sensors announce their availability through a DAQ/LabView interface.
3	A python-based Middleware script discovers new data files being published by LabView and calls the DTUpdate() to update a dynamic pandas dataframe based repository in the Digital Twin.
4	State Estimation Module calls the DTQuery() function in the Digital Twin to request configuration information about the sorbent bed, the available sensors in particular.
5	State Estimation Module selects the sensors that it can use to estimate internal states of the sorbent bed and asks for data.
6	The Digital Twin produces the logs and hands them over to the module.
7	State Estimation Module calls the DTModel() function to request physics-based model results based on model parameters it provides to the Digital Twin.
8	The Digital Twin processes the simulation model as requested and returns the results to the module.
9	The module then processes its accumulated inputs through an Extended Kalman Filter and produces an estimate of the current states-of-interest of the sorbent bed.
10	The current state estimates of the sorbent bed are communicated to the Digital Twin and stored, along with metadata about these estimates (e.g., which agent/version produced them, what uncertainty was communicated, the date/time of the estimate, etc.).

Results

Upon conclusion of the process outlined above in Table 6, the State Estimation Module will have updated the Digital Twin with estimated states at a series of nodes internal to the sorbent bed. Figure 20 shows a printout of the estimated $\%CO_2$ at the midpoint of the sorbent bed.

	timeStamp	data	uncertainty	compName	compID	sensorState	Parameter	dataUnits	dataSource	dataGenTime
0	18000	0.268040433	0	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021
1	18000.5	0.268995248	0.0333333333333333	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021
2	18001	0.26996786	0.0499988902370205	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021
3	18001.5	0.269165892	0.0499992601253325	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021
4	18002	0.266979744	0.0499992601307895	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021
5	18002.5	0.269038948	0.0499992601303347	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021
6	18003	0.265825203	0.0499992601303347	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021
7	18003.5	0.26520257	0.0499992601303347	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021
8	18004	0.267756812	0.0499992601303347	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021
9	18004.5	0.267486843	0.0499992601303347	SorbentBedMidpoint	001	State	CO2	%	EstimationModule	01/03/2021

Figure 20: Printout of resulting dataframe after update of the Digital Twin by the State Estimation Module. Each data point is tagged with the appropriate metadata. This data represents the %CO2 at the sorbent bed midpoint.

The state estimation process was continued for an entire adsorption cycle. The results of the study are shown in Figure 21. Visible in this surface plot is immediate decrease is CO_2 as the gas stream enters the sorbent bed and CO_2 molecules begin to adhere to the zeolite pellets. As time passes, the CO_2 storage capacity of the zeolite pellets is approached and $\%CO_2$ in the gas stream passing through the sorbent bed begins to rise until the sorbent bed cannot hold any more CO_2. At this point the system would switch from adsorption to desorption to begin flushing CO_2 from the system.

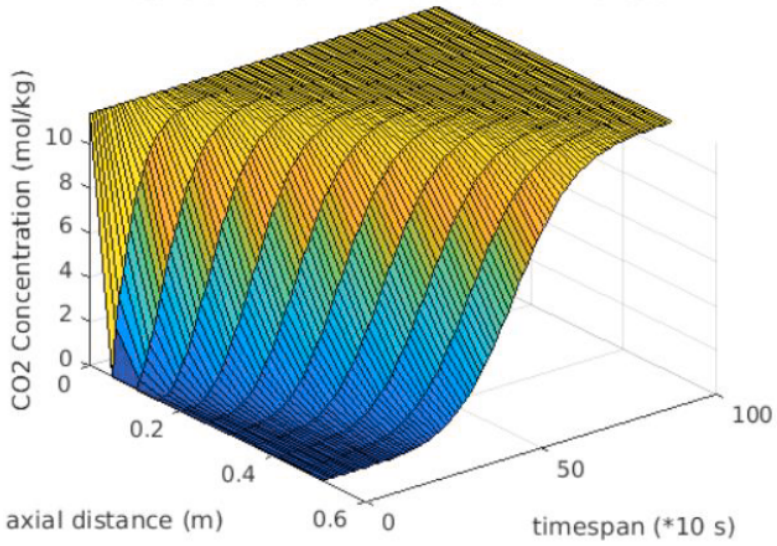

Figure 21: Surface plot showing estimated $\%CO_2$ as a function of axial distance along the length of the sorbent bed and time. This plot is for an entire adsorption cycle. Image Credit: Monica Torralba *(Torralba, et al. 2022)*

The model used here was verified by comparing the STEVE experimental data with the model-simulated data at the outlet of the sorbent bed as seen in Figure 22. The estimation error was plotted in Figure 23 to determine the accuracy of the estimation algorithm.

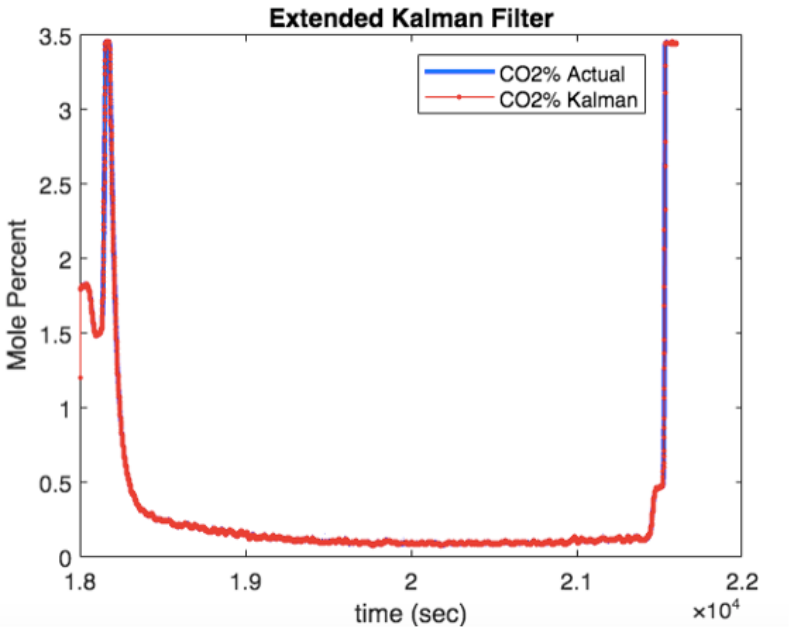

Figure 22: Comparison of Actual Data and Kalman Filter Estimation of $\%CO_2$ at the outlet of the sorbent bed. Showing CO_2 Mole Percent as a function of time. Image Credit: Monica Torralba *(Torralba, et al. 2022)*

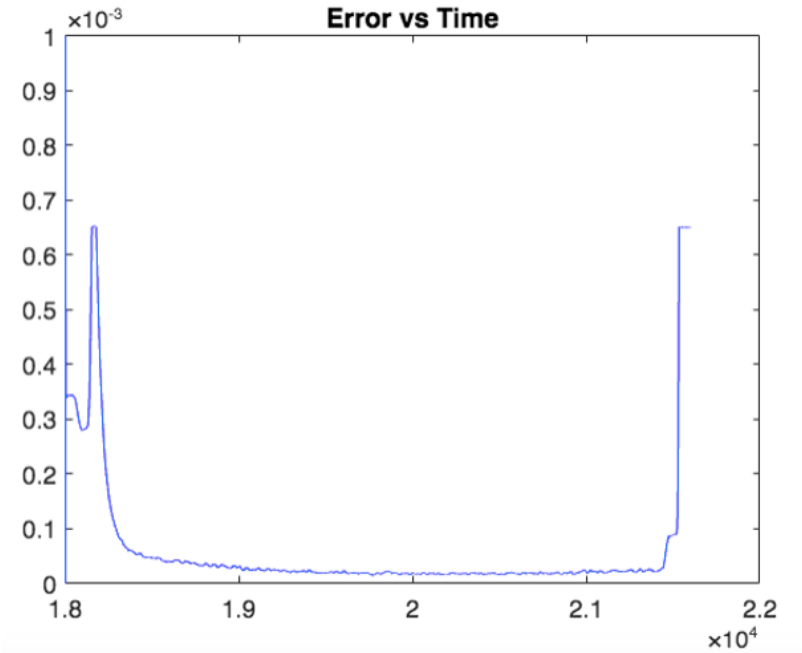

Figure 23: Estimation error for the model and estimation algorithm as a function of time. This error is for the $\%CO_2$ in the airflow at the outlet of the sorbent bed. Image Credit: Monica Torralba *(Torralba, et al. 2022)*

Conclusion

This work was intended to study the utility of Digital Twin technology as a tool to support heightened vehicle autonomy and safety in the next generation of deep-space human habitats. Chapter 1 introduced the concept of a Digital Twin. An abstraction of a Digital Twin Framework was presented with the intention of molding it into a concrete framework as more is learned about what is needed to support deep-space habitat operations. Since a Digital Twin can have generally limitless components, it must be scoped by the needs of its intended users and use-cases. To this avail, Chapter 2 developed detailed use-cases specific to the needs of deep-space missions. The first five use-cases focused largely on Integrated System Health Management scenarios which built on each other to become increasingly complex. They described scenarios in which a Digital Twin would act as a tool in ISHM and decision-making processes. The last two use-cases described scenarios in which a Digital Twin could be useful in taking action on those ISHM decisions. Particularly in supporting robotic M&R tasks.

Chapter 3 used the knowledge gained in Chapter 2 to propose a Digital Twin Framework that could support deep-space habitat operations. Engineering requirements were established for the proposed Digital Twin and supporting infrastructure for the broader Digital Twin Framework. Chapter 4 provided a simple practical demonstration of a Digital Twin based on the System State Estimation use-case. It was shown how a Digital Twin can interface with real hardware and external software modules. The utility of a Digital Twin as a centralized location for system information and modeling was demonstrated.

A Digital Twin is not a stand-alone solution. It is a piece of a broader software package that includes external agents, ISHM algorithms, and robotic agents that interface with it. That broader software package, as a whole, must be capable of performing the job of Mission Control if it is not available for reasons discussed in Chapter 1. If Mission Control is available, that software package should be a tool to augment their work, making it easier and more efficient. Within that broader ecosystem of software, the Digital Twin is acting as a centralized location for information and models that are normally disparate and disconnected. Across the board enforcement of conformity for models and information sources is not something that is currently done but is absolutely necessary to enable the level of vehicle autonomy that is needed to keep people safe in deep space. In this regard, a Digital Twin is not just applicable to NASA's deep-space human exploration goals but is actually an enabling technology. Of course, we can send people into space, but doing so and getting them home safely is the real challenge. No one can ever guarantee that nothing will go wrong, but Digital Twin technology will be a powerful tool if something does.

This work primarily investigated the nature of the interaction between a Digital Twin and its user space. The inner workings of that user space, specifically the algorithms used by software agents, was out of scope and not defined here. Some of the required software elements in the ISHM space have existing solutions mentioned in Chapter 1. For those that don't, the HOME NASA STRI which funded this work, is developing solutions for many of them. One corner of the room that needs further illumination are algorithms that support the proposed What-If Engine (WIE). In Chapter 2, it was proposed that the WIE be able to autonomously develop scenarios representing possible future states of the system. The process for developing these scenarios needs further investigation. It has been suggested that downward counterfactual reasoning may be a good starting point for this process, but these methods need more attention to formalize. There may be other advantageous methods to investigate as well.

The proposed hierarchical computational architecture needs significant attention, specifically the interaction between Digital Twins and supporting software elements at different hierarchical levels. Exactly what processes are handled at each level needs to be thoroughly defined. Processes should be distributed in a way that maximizes computational efficiency while keeping in mind that passing computation requests and information between hierarchical levels has its own cost. The question of which software entity decides the criticality of various tasks and allocates computational resources across the software ecosystem needs to be answered, and how that entity does those things. What information models should be stored in Digital Twins at different levels as well as the appropriate scope and resolution of simulation models also needs to be thoroughly defined.

Work needs to be done mapping out communication and data exchange protocols between all elements of the Digital Twin Framework. There is an open question of how information formatting is addressed both going into and out of the Digital Twin. Users of the Digital Twin, specifically software agents, will likely have been developed by different software development teams from around the world with differing information formatting standards. The choice needs to be made whether to issue a singular information formatting standard to all participating parties or if there is some formatting service built into the Digital Twin that can handle all the various formatting issues. The sooner this decision is made, the better. An early consensus on formatting will streamline development and will decrease the chances of software failures in the future.

Another important element of the proposed Digital Twin that needs more thought and development is the Model Enforcement Conformity Driver (MCED). The MCED is responsible for

monitoring the delta between information models and simulation results, however it may not always be apparent which one is correct. In the case of sensor drift for example, the simulation model in question may be predicting the behavior of the Physical Asset correctly, but sensor values are not matching with model predictions. The MCED should be able to identify when this is occurring. This work also does not address how parameters are optimized and subsequently updated within the Digital Twin.

The realization of the Digital Twin proposed in this work and the fulfillment of the identified requirements will require significant research into the topics outlined above. Additional required research topics include reduced-order system-of-systems modeling, cross-scale coupling of models, integrated model verification and validation, and semantic interoperability that allows clear communication of any data, information, or query between the Digital Twin and its users.

Finally, we leave you with some high-hanging fruit for future researchers, the consideration of the most important Physical Asset onboard the spacecraft, the human crew. If the highly coupled systems of a deep-space habitat represent a complex modeling problem, the complexity and dynamic nature of the human body is even more daunting. (Laubenbacher, et al. 2022) has suggested that a medically transformative Digital Twin would require a large collaboration between government, commercial, and academic organizations. A barrier to implementing such a Digital Twin is the enormous task of calibrating model parameters to an individual. From tissue properties to individual cellular metabolism to overall structure, the variation from person to person is enough to make the individualization of a medical Digital Twin a potentially time consuming and expensive task. To make things worse, just like all things biological, these values are in a constant state of change.

If collaborative parties can figure out how to integrate all of the models of human body systems that have been developed throughout the academic world and quickly, cheaply, and actively individualize model parameters, a complete patient-specific Digital Twin could have boundless benefits. The health of a human crew on a long, deep-space mission would benefit from the situational awareness of internal states of their own bodies. A medical Digital Twin could allow for nutrition, exercise, rest, and daily task working optimization that could lead to a healthier, better-feeling, and more productive crew member. Of-course, the entire field of terrestrial medicine would benefit immensely. We envision a patient's traditional medical records to be augmented with a complete Digital Twin that would stay with a patient for life, informing everything from individualized surgery preparation, medication, and therapy optimization. The possibilities are potentially limitless and certainly worth further study by future researchers.

Works Cited

Badger, Julia M., and Jeremy Frank. 2018. "Spacecraft Dormancy Operational Design for a Crewed Martian Reference Mission." NASA/TM-2018-219965.

Badger, Julia M., Philip Strawser, and Charles Claunch. 2019. "A Distributed Hierarchical Framework for Autonomous Spacecraft Control." *2019 IEEE Aerospace Conference.* Big Sky, MT.

Baidya, S., S. K. Das, M. H. Uddin, C. Kosek, and C. Summers. 2022. "Digital Twin in Safety-Critical Robotics Applications: Opportunities and Challenges." *41st IEEE International Performance Computing and Communications Conference.*

Bellinger, N., E. J. Tuegel, A. R. Ingraffea, T. G. Eason, and S. M. Spottswood. 2011. "Reengineering aircraft structural life prediction using a digital twin." *International Journal of Aerospace Engineering.*

BJELLAND, Ø., B. RASHEED, H. G. SCHAATHUN, M. D. PEDERSEN, M. STEINERT, A. I. HELLEVIK, and R. T. BYE. 2022. "Toward a Digital Twin for Arthroscopic Knee Surgery: A Systematic Review." *IEEE Access.*

Carter, Layne, Jennifer Pruitt, Christopher A. Brown, Jesse Bazley, Daniel Gazda, Ryan Schaezler, and Lyndsey Bankers. July 2016. "Status of ISS Water Management and Recovery." *46th International Conference on Environmental Systems.* Vienna, Austria.

Colombano, S., L. Spirkovska, V. Baskaran, G. Aaseng, R. S. McCann, J. Ossenfort, I. Smith, D. L. Iverson, and M. Schwabacher. 2013. "A system for fault management and fault consequences analysis for NASA's Deep Space Habitat." *AIAA SPACE 2013 Conference and Exposition.* San Diego, CA.

Eshima, Samuel, and Madisen Purifoy-Frie. 2021. *STEVE User Guide Rev. 2.0 5/24/2021.* HOME NASA SRTI.

Figueroa, Fernando, Lauren Underwood, Jonathan Morris, Mark Walker, and Rane Brown. n.d. "Risk-Reduction Autonomy Implementation to Enable NASA Artemis Missions." *2022 IEEE Aerospace Conference (AERO).*

Figueroa, Fernando, Lauren Underwood, Mark G. Walker, and Jonathan Morris. 2019. "NASA Platform for Autonomous Systems (NPAS)." *AIAA SciTech Forum.* San Diego, California.

Fong, T. W., J. D. Frank, J. M. Badger, I. A. Nesnas, and M. S. Feary. 2018. "Autonomous System Taxonomy." May. Accessed July 2022.

FULLER, A., Z. Fan, C. Day, and C. Barlow. 2020. "Digital Twin: Enabling Technologies, Challenges and Open Research." *IEEE Access.*

Gratius, Nicolas, Zhichen Wang, Min Hwang, Yu Hou, Mario Berges, Burcu Akinci, Annika Rollock, and Cory A. George. 2023. *Digital twin technologies for self-aware and self-sufficient environmental control and life support systems: A literature review.* Anticipated Publisher: Journal of Aerospace Information Systems.

Guivarch, D., E. Mermoz, Y. Marino, and M. Sartor. 2019. "Creation of helicopter dynamic systems digital twin using multibody simulations." *CIRP Annals - Manufacturing Technology.*

Harris, Danny W., Paul D. Kessler, Tiffany M. Nickens, Andrew J. Choate, Bryce L. Horvath, Matthew A. Simon, and Chel Stromgren. 2022. "Moon to Mars (M2M) Habitation Considerations A Snap Shot As of January 2022." *NASA/TM-20220000524.*

HOME. 5-31-2022. "HOME_STRI_Year 3 Annual Report."

Huang, Sihan, Guoxin Wang, and Yan Yan. 2020. "Building blocks for digital twin of reconfigurable machine tools from design perspective." *International Journal of Production Research.*

Hwang, Min Young, Burcu Akinci, and Mario Berges. 2022. "Updating Subsystem-Level Fault-Symptom Relationships for Temperature and Humidity Control Systems with Redundant Functions." *73rd International Astronautical Congress (IAC).* Paris, France.

Kapteyn, Michael G. 2021. *Mathematical and Computational Foundations to Enable Predictive Digital Twins at Scale.* Department of Aeronautics and Astronautics, Massachusets Institute of Technology.

Klaus, D., A. Rollock, P. Pischulti, and S. Zaccarine. Rev September 30, 2020. *Critical Terms and Definitions.* Habitats Optimized for Missions of Exploration (HOME).

Laubenbacher, R., A. Niarakis, T. Helikar, G. An, B. Shapiro, R. S. Malik-Sheriff, T. J. Sego, A. Knapp, P. Macklin, and J. A. Glazier. 2022. "Building digital twins of the human immune system: toward a roadmap." *npj Digital Medicine.*

Liao, M., G. Renaud, and Y. Bombardier. 2020. "Airframe digital twin technology adaptability assessment and technology demonstration." *Engineering Fracture Mechanics.*

Mohammadi, N., and J. E. Taylor. 2017. "Smart city digital twins." *IEEE Symposium Series on Computational Intelligence.*

NASA. 2022. *NASA Software Engineering Requirements.* NPR 7150.2D.

NASA. 2015. "NASA Technology Roadmaps, TA 4: Robotics and Autonomous Systems."

NASA. 2020. *Software Assurance and Software Safety Standard.* NASA-STD-8739.8A.

Pischulti, P, S Zaccarine, A Rollock, and D Klaus. May 11, 2020. *HOME (orbital) - Design Reference Mission (DRM).* University of Colorado, Boulder.

Sacks, R., I. Brilakis, E. Pikas, H. S. Xie, and M. Girolami. 2020. "Construction with digital twin information systems." *Data-Centric Engineering* (Cambridge University Press) 1 : e14.

Shafto, Mike, Mike Conroy, Rich Doyle, Ed Glaessgen, Chris Kemp, Jacqueline LeMoigne, and Lui Wang. 2012. *Modeling, Simulation, Information, Technology & Processing Roadmap Technology Area 11.* National Aeronautics and Space Administration.

Shao, Guodong, and Moneer Helu. 2020. "Framework for a digital twin in manufacturing: Scope and requirements." *Manufacturing Letters.*

Torralba, Monica G., Cory A. George, Stephen K. Robinson, Samuel P. Eshima, and James A. Nabity. 2022. "Estimation of System States for Non-Measured Parameters and Integration with a Digital Twin Framework to Boost Spacecraft Autonomy and Awareness." *51st International Conference on Environmental Systems.* St. Paul, Minnesota.

Vering, Christian, Philipp Mehrfeld, Markus Nürenberg, Daniel Coakley, Moritz Lauster, and Dirk Müller. 2019. "Unlocking Potentials of Building Energy Systems' Operational Efficiency: Application of Digital Twin Design for HVAC systems." *Proceedings of the 16th IBPSA Conference.* Rome, Italy.

Williamson, Jill P., Layne Carter, Jimmy Hill, Davey Jones, Danielle Morris, and Rex Graves. July 2019. "Upgrades to the International Space Station Urine Processor Assembly." Boston, Massachusetts: 49th International Conference on Environmental Systems.

Woo, Gordon. 2019. "Downward Counterfactual Search for Extreme Events." *Frontiers in Earth Science.*

Xie, X., Q. Lu, A. K. Parlikad, and J. M. Schooling. 2020. "Digital Twin Enabled Asset Anomaly Detection for Building Facility Management." *IFAC-PapersOnLine 53 (3)* 380–385.

Xu, Jiuping, and Lei Xu. 2017. *Integrated System Health Management Perspectives on Systems Engineering Techniques.* Elsevier Inc.