Artificial Intelligence Agents to Support Data Mining for SoS Modeling of Space Systems Design

Cesare Guariniello
Purdue University
School of Aeronautics and Astronautics
701 W. Stadium Ave
West Lafayette, IN
cguarini@purdue.edu

Thomas B. Marsh
ai-one, Inc.
American Headquarters
5666 La Jolla blvd #104
La Jolla, CA
tm@ai-one.com

Ron Porter
Von Braun Center for
Science and Innovation
139 High Coach Way
Madison, AL
IDEA.vcsi@yahoo.com

Christopher Crumbly
University of Tennessee
Space Institute
411 B.H. Goethert Pkwy
Tullahoma, TN
ccrumbly@utsi.edu

Daniel A. DeLaurentis
Purdue University
School of Aeronautics and Astronautics
701 W. Stadium Ave
West Lafayette, IN
ddelaure@purdue.edu

Abstract—**The complex and multidisciplinary nature of space systems and mission architectures is especially evident in early stage of design and architecting, where systems stakeholders have to keep into account all the aspects of a project, including alternatives, cost, risk, and schedule and evaluate various potentially conflicting metrics with a high level of uncertainty. Though aerospace engineering is a relatively young discipline, stakeholders in the field can rely on a vast body of knowledge and good practices for space systems design and architecting of space missions. These guidelines have been identified and refined over the years. However, the increase in size and complexity of applications in the aerospace discipline highlighted some gaps in this approach: first, the amount of available information is now very large and originates from multiple sources, often with diverse representations, and useful data for trade space analysis or analysis of all potential alternatives can be easily overlooked; second, the variety and complexity of the systems involved and of the different domains to be kept into account can generate unexpected interactions that cannot be easily identified; third, continuous advancements in the field of aerospace resulted in the development of new approaches and methodologies, for which a common knowledge database is not existing yet, thus requiring substantial effort upfront. To address these gaps and support both decision making in early stage of space systems design and increased automation in extraction of necessary data to feed working groups and analytical methodologies, we propose the training and use of Artificial Intelligence agents. These agents can be trained to recognize not only information coming from standardized representations, for example Model Based Systems Engineering diagrams, but also descriptions of systems and functionalities in plain English. This capability allows each agent to quantify the relevance of publications and documents to the query for which it is trained. At the same time, each agent can recognize potentially useful information in documents which are only loosely connected to the systems or functionalities on which the agent has been trained, and which would possibly be overlooked in a traditional literature review. The search for pertinent sources can be further refined using keywords, that let the user specify more details about the systems or functionality of interest, based on the intended use of the data. In this work we illustrate the use of Artificial Intelligent agents to sort space habitat subsystems into NASA Technology Roadmaps categories and to identify relevant sources of data for these subsystems. We demonstrate how the agents can support the retrieval of complex information required to feed existing System-of-Systems analytic tools and discuss challenges of this approach and future steps.**

978-1-7281-2734-7/20/$31.00 ©2020 IEEE

ACRONYMS

AI	Artificial Intelligence
AWB	Analytic Work Bench
CDH	Command and Data Handling
COD	Criticality of Dependency
DIM	Disruption Impact Matrix
DoD	Department of Defense
ECLSS	Environmental Control and Life Support Systems
IOD	Impact of Dependency
MBSE	Model-Based Systems Engineering
MMOD	Micro Meteoroid and Orbital Debris
NLP	Natural Language Programming
NTRS	NASA Technical Reports Server
PERT	Project Evaluation and Review Technique
SE	Self-Effectiveness
SERC	Systems Engineering Research Center
SDDA	Systems Developmental Dependency Analysis
SME	Subject Matter Experts
SOD	Strength of Dependency
SODA	Systems Operational Dependency Analysis
SoS	System-of-Systems
TABS	Technology Area Breakdown Structure
TRL	Technology Readiness Level

TABLE OF CONTENTS

1. INTRODUCTION 1
2. SYSTEM-OF-SYSTEMS ANALYSIS AND SYNTHESIS: ANALYTIC WORK BENCH AND PREVIOUS STEPS .. 2
3. ARTIFICIAL INTELLIGENCE AGENTS 3
4. CASE STUDY: THE NASA GATEWAY HABITAT 4
5. CONCLUSIONS AND FUTURE WORK 8
ACKNOWLEDGMENTS 10
REFERENCES 10
BIOGRAPHY 10

1. INTRODUCTION

Architecting a complex space mission involves various multidisciplinary decisions variables [1] and variations along any of these decision variables results in a distinct architecture [2], where the system roles and their interactions with other systems could vary drastically and cause substantial fluctuations in the behavior of the systems and the outcome of the mission. This makes space systems design a typical System-of-Systems (SoS) problem. For space missions, given the extensive design space and a large number of SoS architectures, it becomes impractical to employ engineering judgment alone to identify the common features of different well performing architectures and therefore to take decisions well-supported by quantitative analysis. The volume, veracity and velocity of research exarcebates the engineering challenge in an area where the stakes could not be higher.

In past applications, our team used a suite of tools to address SoS aspects of space systems architecture [3], [4], [5]. The suite of tools, called SoS Analytic Work Bench (AWB) and described in section 2, has been used in a variety of research sectors, including Cybersecurity [6] and Global Navigation Satellites Systems [7]. However, collaboration with Subject Matter Experts (SME) from NASA provided a useful testbed for analysis of space mission which resulted in substantial improvement of the usefulness and usability of the AWB [8]. Past improvement focused on the utilization of Model-Based Systems Engineering (MBSE) representations to facilitate input and output processes, and on support to SME in the phase of modeling the systems behavior through tools in the AWB. However, due to the size and complexity of space system design and space mission architectures and to the relative novelty of the tools in the AWB, the data collection to gather the necessary information for the AWB tools is still a bottleneck in the process.

To address this problem and support the data mining requirement, we propose the implementation and utilization of Artificial Intelligence (AI) agents to power a specialized space architecture database. These agents, once trained on case studies, can navigate repositories of documents and publications to find the necessary data to run SoS analysis with tools in the AWB. Figure 1 shows the various parts of this research application

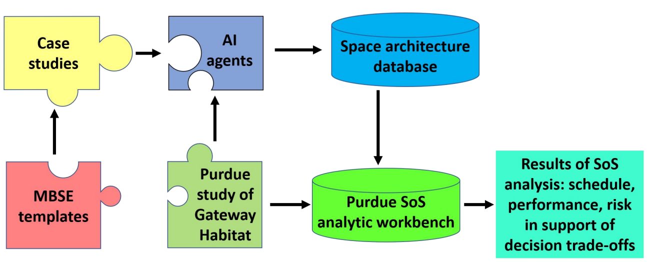


Figure 1. Combined application of Artificial Intelligence and SoS Analytic Work Bench. Based on case studies, with possible information in MBSE format, the AI agents build a database that feeds tools in the AWB for SoS analysis.

Though SMEs are currently still heavily involved in the process of selecting the appropriate sources and extracting the necessary information, the use of AI agents not only accelerates the process, but can also identify valuable sources that have only a loose connection with the functionality or the system under study, and would therefore likely be overlooked without the support of the agents. The flexible design of the AI agents accommodates training information provided by the user in different formats, including plain English description, text examples and keywords and is an important step towards increased automation of the SoS analysis and synthesis process. To demonstrate the combined use of AI agents and AWB tools, this paper focuses on the analysis of the habitation portion of NASA Lunar Gateway.

2. System-of-Systems Analysis and Synthesis: Analytic Work Bench and Previous Steps

As mentioned in section 1, the collection of systems that constitute a SoS can exhibit traits of operational and managerial independence, geographic distribution, emergent behavior, and evolutionary development [9], [10]. Due to their very nature, SoS cannot always be analyzed with conventional Systems Engineering methodology. Research in the SoS field at Purdue University addressed multiple aspects of SoS, in particular the dynamic behavior due to the interactions among constituent systems. The result was the creation of an Analytic Work Bench (AWB), developed within research projects of the Systems Engineering Research Center (SERC) to meet the needs of the US Department of Defense (DoD) for new methodologies to be used for analysis and synthesis of SoS architectures [11]. The AWB is a suite of methods and tools that can be used to achieve a top-level systemic assessment touching different aspects of SoS engineering [12]. The suite includes some tools that can assess the developmental risks and uncertainty in time and resources, and policy contextual questions [13], [14] and other tools and methodologies that provide information on the operational aspects of complex architectures [15], [16]. In general, the tools in the AWB are oriented to the simulation and analysis of SoS, with the goal of providing holistic assessment of the complex architectures associated with SoS problems. Results of this assessment take the form of metrics that quantify properties of the SoS as a whole, accounting for the interactions between the systems and possible emergent behavior due to these interactions. The research presented in this paper focused on the support provided by AI agents to the retrieval of useful information for two methods in the AWB: Systems Developmental Dependency Analysis (SDDA) and Systems Operational Dependency Analysis (SODA).

Systems Operational Dependency Analysis

SODA methodology, developed in part based on Functional Dependency Network Analysis [17], [18], addresses the operational domain of a SoS, by providing analysis of the impact of dependencies between constituent systems on the propagation of the effect of disruptions. In SODA, a parametric model of system behavior is combined with a network representation for the system architecture. Figure 2 shows an example of this representation for the high-level systems of a Lunar Gateway habitation module, where the nodes are systems within the architecture, and the edges are operational dependencies between the systems.

In SODA, a small set of parameters is used to produce a simple model of the dependencies between each system. These parameters represent aspects of the dependency of the operability of a system on the operability of another systems [16]. The Strength of Dependency (SOD) represents a linearized operational dependency between systems in the case of small disruptions. The Criticality of Dependency (COD) represents the loss of operability due to major disruptions. The Impact of Dependency (IOD) models the boundary between the small disruption regime and the major disruption regime. Figure 3 shows the SODA piecewise linear model of the dependency between two systems. Based on the parameters of the model, SODA can quantify the cascading effect of disruptions in the architecture and constitutes a quantitative method of risk

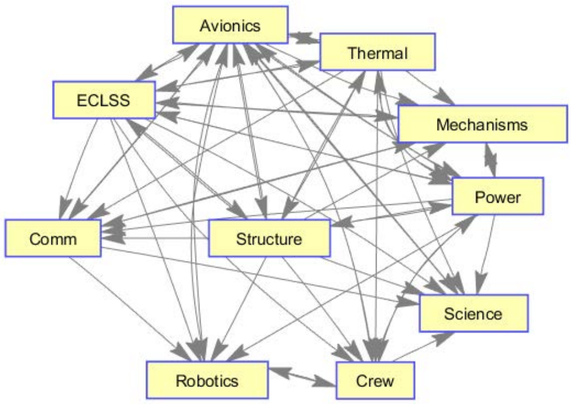


Figure 2. Operational dependency network for a habitation module.

analysis which can be used to expand the traditional risk matrix. The algorithm can also model partial failures, both deterministic and stochastic, and multiple paths of propagation within the model. SODA thus provides early-stage feedback for the architecture's design, reducing the amount of simulation and other verification methods required to ensure mission feasibility and to identify criticalities and areas of potential emergent behavior.

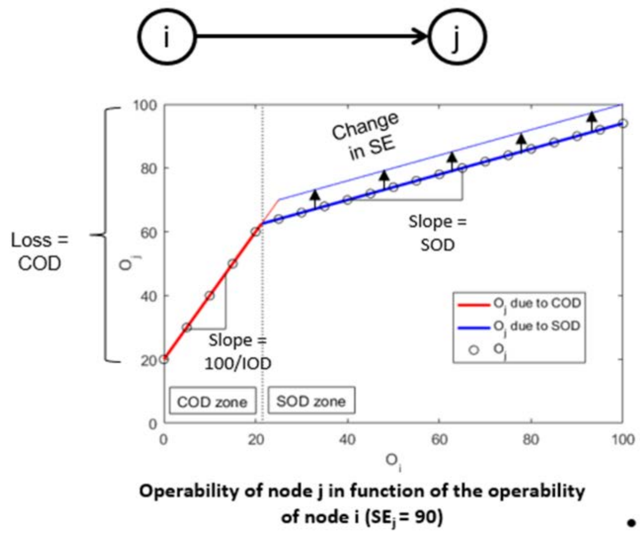


Figure 3. SODA piecewise linear model of dependency of the operability of system *j* on the operability of system *i*.

Systems Developmental Dependency Analysis

Sister to SODA, but applied to the developmental domain, SDDA provides a parametric model of the interactions between constituent systems of a SoS for what concerns development and schedule. The Strength of Dependency (SOD) in SDDA model evaluates the fraction of development time of a system that is dependent on inputs by other systems. With this parameters, SDDA can model partial developmental dependencies and partial parallel development. The model also account for the punctuality of a system development, that is how much the system is following the expected schedule. The Criticality of Dependency (COD) models the level of punctuality below which partial parallel development of systems is not acceptable, and the dependency between systems become analogous to PERT networks, where a task needs to be fully completed before following tasks can begin. Figure 4 shows the model of developmental dependency between two systems. The algorithm is used to evaluate the impact of delays in the development of individual systems on the schedule of the whole SoS. A stochastic model, based for example on Technology Readiness Level (TRL), can be used to evaluate uncertainty on the schedule and the most critical technologies to be developed early in order to guarantee timely completion of the SoS development.

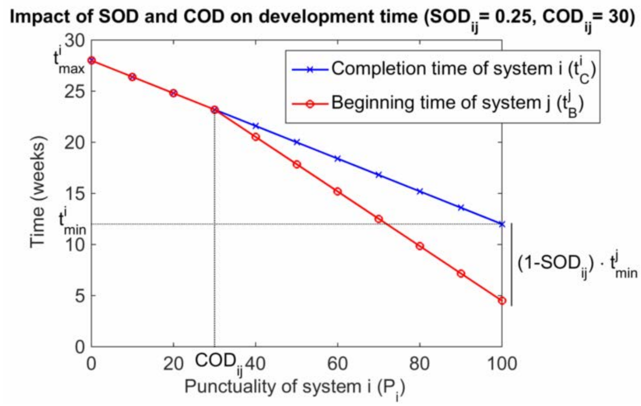


Figure 4. Completion time of system *i* and beginning time of system *j* in function of the parameters of the developmental dependency between the two systems. Due to partial dependency, system *j* can begin its development before completion of system *i*, unless *i* is critically late.

3. Artificial Intelligence Agents

NASA Technology Roadmaps and AI Agents

The requirements for research support for the AWB tools demanded a new approach. To ensure the quality of the research collected, we focused on content curated by NASA and stored in the NASA Technical Reports Server (NTRS) repository. Traditional Boolean search relies heavily on keywords and the accuracy of tagging provided by the author of the articles. This approach leads to results that miss articles that are incorrectly tagged or that containing relevant data but were published against an unrelated primary topic. Boolean results are also binary in that a relevant article that does not contain the right keyword is completely missed and there is no approach that enables the researcher to be confident they have found all the relevant articles. This problem is especially acute for space research, as many of the topics have not been fully discovered/invented yet and hence do not have a known keyword for searches. Similarly, engineering terms evolve and today's researcher may not be aware of previously used terms that would lead to relevant articles. Boolean search relies on the user to have prior knowledge of key terms to use to initiate a search. For research on a low TRL technology, the solution may not be mature enough to have a keyword or the keyword may be difficult to find. For a rapidly evolving technology, the keywords used in earlier research may be obsolete rendering that tagging marginally useful. Additionally, the best search may be the result of a combination or pattern of relatively common terms. While repeated guessing at search terms and combinations can be time consuming, our approach builds a search agent automatically from multiple examples, doing the work for the researcher and accelerating the process.

To solve this problem, we developed a new approach to building an enriched database of space architectures, at the core of which is a new AI technology for topic discovery. These AI topic agents, originally developed by ai-one inc, were used along with document metadata and entity extraction to classify over 60,000 research papers in the NTRS repository. The agents trained for this project were based on the 354 Technology Area Breakdown Structure (TABS) agents from earlier work. Trained directly from the NASA 2015 Technology Roadmaps [19], they were consistent with NASA's existing ontology.

For this project, Purdue and other teams submitted descriptions, grouped by system, of each subsystem in their architecture. The descriptions included a paragraph of text, possible keywords, and reference sections to TABS if available. The initial inputs from the Purdue team for the habitat portion of NASA Lunar Gateway consisted of 27 subsystems grouped into 10 systems. An AI agent was created for each, which was then used to deliver a set of relevant articles for that subsystem.

Agent Training Process

We chose not to use unsupervised machine learning to provide the user with control over an intuitive flexible tool that once trained would deliver consistent results as part of the processing pipeline. The supervised learning process used for this Agent training consisted of supplying the agent with keywords, descriptions of each subsystem and samples of the topic from the NASA content, including sentences as well as paragraphs. Once deployed, each of the agents analyzes millions of paragraphs in the 60,000 papers and returns a similarity score for each, depending on how closely the paragraph reflects the topic. The technology used abstracts or generalizes the concept being scored from the samples, words and their patterns in language. Specifically, this technology is focused on the detection of concepts, topics or themes at the paragraph and/or sentence level as a means to accurately define the context(s) of the text before performing more specific analyses such as sentiment, grammar, entity extraction, etc. The technology for concept detection is a derivative from work in neural networks, Natural Language Programming (NLP) and computational linguistics. In training, the agents learn words and patterns, stored as an array that indicate the concept or topic. The agent training process also allows the user to boost the score if any of a list of specific unambiguous keywords or regex expressions are found in the text. These additional word patterns can pick up topics that would have otherwise been missed with a Boolean search method. Agents are tested and retrained periodically to accommodate relevant new terms and language patterns as language evolves, especially important in the fast evolving and state-of-the-art space domain. The array for an agent is then compared with the paragraph arrays in each paper and scored for similarity. The use of a normalized similarity score instead of a binary classifier provides a resulting system which is more flexible across large variances in language. Our technology generates a similarity score from the comparison of an agent array with the array built for each paragraph of target text. In our application the cutoff value is a parameter adjustable by the user through an interactive graphical user interface. Our research has found concepts or topics most accurately expressed at the paragraph level. A hit for a document can come from the highest single paragraph score or aggregating the similarity scores for all the paragraphs in the document. Ranking is also adjusted through the use of extracted entity and document metadata. All of this is available to and can be manipulated by the user. Additionally, our algorithms and those used by other researchers can be incorporated into the processing pipeline and those attributes added to the database with each paragraph, so multiple attributes can be compared and/or extended by our users or other researchers to provide a superior tool for research. The similarity scores returned by each agent for each paragraph are then consolidated into a final classification of topics for a given paper. In addition to the similarity score calculated by the agent, the user can provide keywords or phrases that if present will boost the overall similarity score. This additional boost can be set from zero to 100%, resulting in an overall score maximum of 2. In this particular study, the boost was set to 50% for all agents, hence the similarity score ranges from 0 to 1.5. While not used to optimize results in this case, the ability to adjust these parameters allow the agents to score different types of text (ex. News vs Research) and return similar relevancy scores as perceived by the user. Critical to our approach is the user's ability to control all of these variables, yielding transparent and repeatable results with no black box in the background. The classification is accomplished by adjusting the cutoff value for the consolidated similarity scores for a paper. If the similarity score is greater than or equal to this cutoff value, then the topic is deemed to be present in the paper. If the similarity score is less than this cutoff value, then the topic is not present in the paragraph/paper. We call this feature the "almost" feature as the user, in combination with other attributes (date, author, organization, etc), can lower the cutoff to see more research or increase it to reduce the size of corpus for research to be downloaded and reviewed. As part of this process in creation/training of the classifiers for each agent, the results of each agent have been tested against a corpus and scored for relevancy, i.e. percentage of false positives and false negatives. Figure 5 illustrates the phases of the AI agents training process.

Potential expansion

The creation of these agent classifiers is a continuous, collaborative effort within the space engineering community to build a complete set of research tools for all the possible space architecture elements. As the collection of agents is built out, engineers seeking research or updates to their research corpus will be able to use the space architecture database to quickly build and populate new models providing program managers with a near real time tools for modeling different scenarios to reduce risk to schedule, scope and cost.

4. Case Study: The NASA Gateway Habitat

Problem Description

The case study models and analyzes operational and developmental dependencies between constituents of the habitation module in the NASA cislunar Gateway. Functional and systemic decomposition has been used both to identify the systems and subsystems in the habitat architecture and to provide description of these components and their functionality for the AI agents training. The team identified 10 systems and 27 subsystems, shown in figure 6. Figure 7 shows an example of the documentation with the description of subsystems and their functionality.

The objective of the problem is to use results from the AI agents search to retrieve useful information that can be fed into tools of the AWB for analysis of SoS features of the habitat architecture. This process requires three steps:

Figure 5. Phases of the AI agents training: implementation of database of abstracts; creation of agents; scoring of the abstracts with agents; presentation of results in Business Intelligence dashboard.

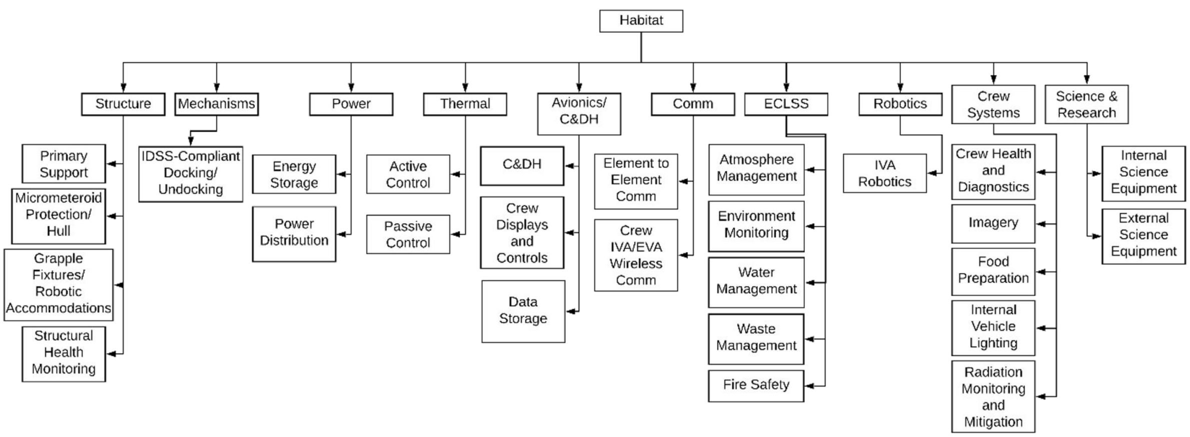


Figure 6. Systems (top row) and subsystems in the NASA Gateway Habitat.

System	Subsystem	Description
	Primary Support Structure	Main load-bearing structure of the spacecraft such as beams and trusses that support the hull and provide attachment points for other systems in the spacecraft.
	Micrometeoroid Protection/Hull	Outer surfaces and hull of the spacecraft that form the barrier between the habitable environment inside the spacecraft and the outside space environment. The hull must protect the crew from environmental hazards like radiation and micrometeroid impacts. The integrity of the hull is essential for maintaining pressure and atmosphere in the spacecraft.
Structures	Grapple	Fixtures on the outer surface of the habitat that allow robots to grapple onto the module. These fixtures allow external robotic arms (like

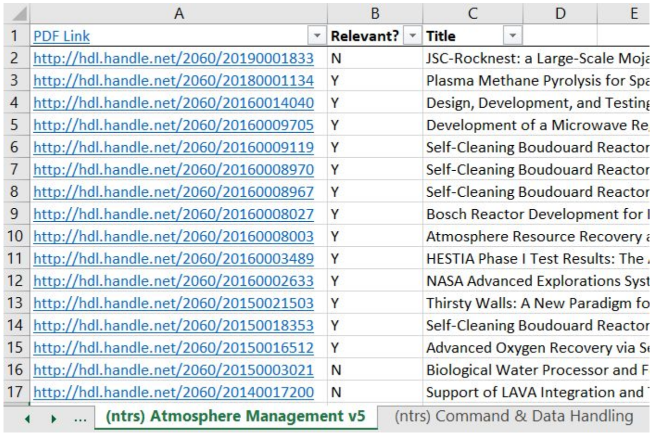
Figure 7. The AI agents have been trained based on description of the subsystems and their functionality.

- Training of the AI agents
- Utilization of the agents to identify relevant sources and extraction of the necessary information
- Analysis of the habitat architecture with AWB tools.

Agent Training and Refinement

The AI agents described in section 3 have been trained using the description of the subsystems provided by the team at Purdue. Tested against the database, the agents have then been refined producing more advanced versions of them. During this process, experts have been queried to support the boosting of specific concepts or keywords, in order to obtain more focused results that can provide the best sources of information to the AWB. Figure 8 shows some of the results of the search performed by version 5 of the AI agent for the Atmosphere Management subsystem. The spreadsheet shows a summary of the information that the user can see in the Business Intelligence dashboard. To improve the agents, 20 literature sources have been evaluated by SMEs for what concerns their relevance to the topic. Following this evaluation, new versions of the agents have been produced which boost or avoid specific keywords and concepts that resulted in irrelevant sources.

During this process, some interesting results provided insight into the capabilities of the AI agents to learn and to interpret concepts. For example, the description for the Grapple Fixture subsystem was written referring to handles that robotic arms can use to dock to and move around the module. However, the AI agent identified sources that describe magnetic mechanism for docking of satellites and robotic arms. These sources are very relevant and match the concept of grappling fixtures, however would have been easily overlooked by users familiar only with handles as grappling fixtures.



	A	B	C	D	E
1	PDF Link	Relevant?	Title		
2	http://hdl.handle.net/2060/20190001833	N	JSC-Rocknest: a Large-Scale Moja		
3	http://hdl.handle.net/2060/20180001134	Y	Plasma Methane Pyrolysis for Spa		
4	http://hdl.handle.net/2060/20160014040	Y	Design, Development, and Testing		
5	http://hdl.handle.net/2060/20160009705	Y	Development of a Microwave Re		
6	http://hdl.handle.net/2060/20160009119	Y	Self-Cleaning Boudouard Reactor		
7	http://hdl.handle.net/2060/20160008970	Y	Self-Cleaning Boudouard Reactor		
8	http://hdl.handle.net/2060/20160008967	Y	Self-Cleaning Boudouard Reactor		
9	http://hdl.handle.net/2060/20160008027	Y	Bosch Reactor Development for I		
10	http://hdl.handle.net/2060/20160008003	Y	Atmosphere Resource Recovery a		
11	http://hdl.handle.net/2060/20160003489	Y	HESTIA Phase I Test Results: The		
12	http://hdl.handle.net/2060/20160002633	Y	NASA Advanced Explorations Syst		
13	http://hdl.handle.net/2060/20150021503	Y	Thirsty Walls: A New Paradigm fo		
14	http://hdl.handle.net/2060/20150018353	Y	Self-Cleaning Boudouard Reactor		
15	http://hdl.handle.net/2060/20150016512	Y	Advanced Oxygen Recovery via S		
16	http://hdl.handle.net/2060/20150003021	N	Biological Water Processor and F		
17	http://hdl.handle.net/2060/20140017200	N	Support of LAVA Integration and		

Figure 8. Spreadsheet showing part of the results of version 5 of the AI agent for Atmosphere Management. The literature sources have been reviewed by Subject Matter Experts, who indicated whether the source is relevant to the topic or not.

Use of the agents to identify data sources

While the spreadsheet used to evaluate the relevance only provides links to the sources, the dashboard yields more information, including dates of publication, authors, and location where the research was conducted. Furthermore, the user can add filters to the set of sources, for example indicating specific keywords. While the long term goal of this research effort is to achieve as much automation of the process as possible, the phase of extracting relevant information from the sources found by the agents has been performed by SMEs. In this phase, the use of filters proved very useful, especially to identify modeling parameters for the SODA tool. Since SODA models the operational dependencies between subsystems, as well as the behavior of the whole architecture when disruptions occur, refinement of the sources using keywords such as "failure" have been used to quickly gather existing information on failure modes and effect of disruptions in subsystems of interest. The results have been integrated with information by SMEs to build SODA models of the subsystems of the Gateway habitat.

For what concerns SDDA, the model requires data about the expected time required to complete the development of systems and technologies, and information about the uncertainty of the expected time. In this case, the relevant features of the AI agents is their training against the NTRS repository. Through the AI agents dashboard, the user can retrieve information about the agent or agents which gave a relevant score to specific sources. Since the agents are associated with NASA TABS, this information relates each sources to its pertaining areas in the TABS structure. Each item in TABS, that is each technology or subsystem, is associated to a current Technical Readiness Level (TRL) and to the TRL required to support specific space missions. This information has been used to build SDDA models of the subsystems of the Gateway habitat. Figure 9 shows the developmental dependencies between subsystems of the habitat, and Table 1 shows the associated TABS. The space architecture database we are proposing to build based on the AI agents will support this process of data retrieval.

Table 1. NASA TABS associated with the habitat subsystems.

Subsystem	*TABS category*
Primary Support Structure	12.1.1, 12.2.6
Micrometeoroid Protection	12.1.4, 12.2.5.2
Grapple Fixtures	4.3.7, 12.3.3.1
Structural Health Monitoring	12.2.3.3
IDSS-Compliant Docking	4.6.3
Energy Storage	3.2.1, 12.1.5.3
Power Distribution	3.3.1, 3.3.5
Passive Thermal Control	12.1.4.5
Active Thermal Control	14.2.1, 14.2.2
Command and Data Handling	11.1.1
Crew Displays and Controls	6.3.4.1, 11.4.7
Data Storage	11.1.1.2
Element to Element Communication	5.3
Crew IVA/EVA Wireless Communication	5.2
Atmosphere Management	6.1.1, 6.4.4.1
Environmental Monitoring	6.4.1
Water Management	6.1.2
Waste Management	6.1.3
Fire Safety	6.4.2, 6.4.4.3
IVA Robotics	4.3.1, 4.3.2, 6.3.4.7
Crew Health and Diagnostics	6.3
Imagery	4.1
Food Preparation	6.1.4.9, 6.1.4.10
Internal Vehicle Lighting	6.3.3.3
Radiation Monitoring and Mitigation	6.5.1, 6.5.3, 6.5.4, 12.1.4.4
Internal Science and Research	8.1, 8.3
External Science and Research	8.1, 8.3

Analysis

The SODA and SDDA models of the habitat, built with the support of the AI agents, have been used to perform different types of analysis aimed at identifying the most critical subsystems and technologies. In the operational domain (SODA), these are the elements that have the highest impact on the whole habitat SoS when disruptions occur. The large amount of information about impact of disruptions accounting for the dependencies between systems can be summarized in a Disruption Impact Matrix (DIM). This matrix shows disrupted subsystems in the rows, with a level of disruption that can be modified by the user. The columns show the impacted subsystems, and the colors in the cell represent the status of the impacted subsystems given the disruption in the failed subsystem. Green is nominal status, yellow is sub-nominal status, and red is disrupted status. Figure 10 shows the DIM for the habitat subsystems. Fire safety, primary structure, MMOD, and power distribution appear to be the most critical subsystems in the operational domain. The user can then obtain more details by studying individual disruptions. More detailed results about SODA analysis of the Gateway habitat can be found in [8].

SDDA analysis focuses on the impact that delays in individual subsystems have on the development schedule of the whole habitat. The results are shown in form of Gantt charts,

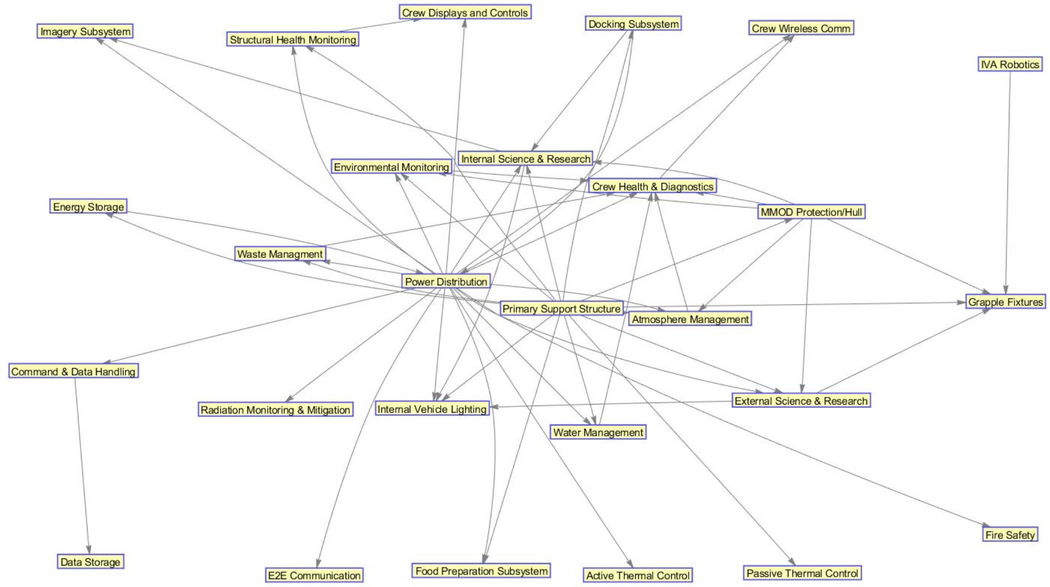


Figure 9. SDDA network for the Gateway habitat, showing the developmental dependencies between the subsystems.

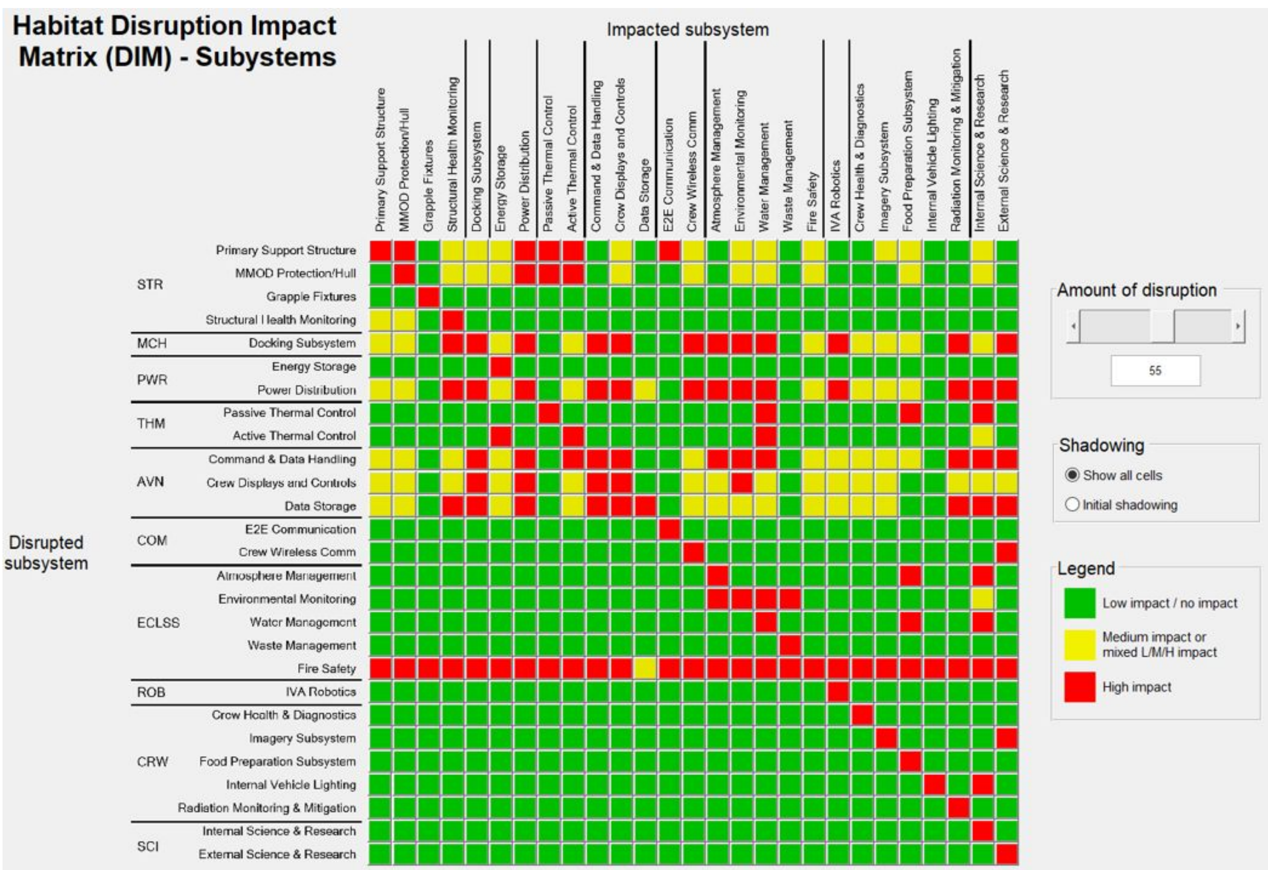


Figure 10. Disruption Impact Matrix for the habitat subsystems. Rows are the disrupted subsystems, columns are the impacted subsystems. The user can modify the amount of disruption to study different impacts in the network.

which show the partial parallel development of the subsystems. SDDA has the capability to automatically reschedule the development times based on the dependencies and on the delays, with large delays causing less parallel development, to represent the lower reliability associated with the delayed development.

Figure 11 shows the Gantt chart for the development and assembly of the Gateway habitat resulting from SDDA analysis in the nominal case, and the same schedule resulting from a version of SDDA that implements more conservative decision making for what concerns the execution of tasks (the conservative approach results in later completion of the entire project, but avoids potential waste of resources due to early beginning of tasks that cannot be completed early because of delays in other tasks).

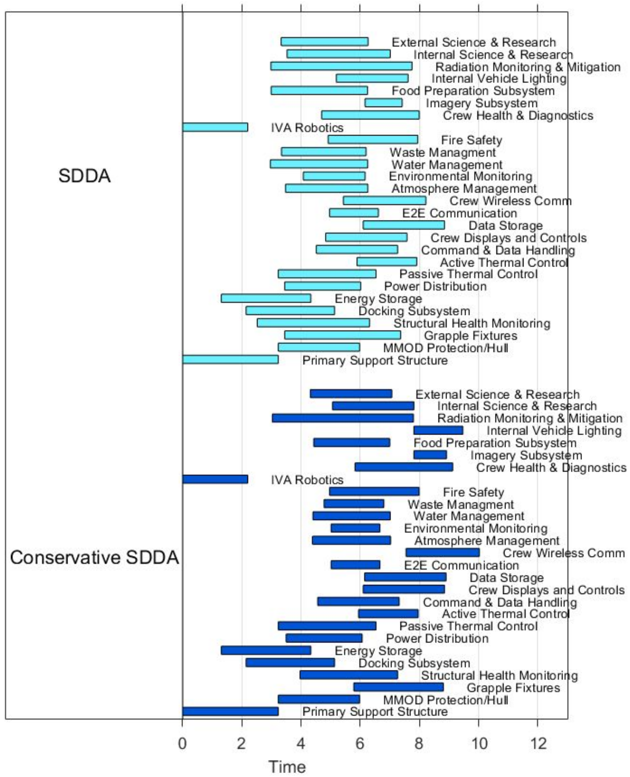


Figure 11. Gantt chart for the development of the habitat according to the SDDA model and to a conservative version of the SDDA model.

To evaluate the impact of delays in individual subsystems on the overall schedule, we performed SDDA analysis where one system at a time has a delay equal to 50% its maximum expected delays. SDDA model quantifies the propagation of this delay to other subsystems, and we assessed the final delay on the completion of the whole SoS development. Figure 12 shows the results of this analysis. The empty rectangles represent the initial delay in the disrupted subsystem, while the colored bars show the final delay in the overall development. Since the model allows for partial parallel development, some of the delays can be totally or partially absorbed. Some initial delays, however, cause an even longer delay on the entire schedule. In the SDDA mode, the Primary Support Structure, the MMOD Protection, and the Power Distribution are the most critical elements, while delays in all of the other subsystems are fully or partially absorbed. The conservative model tends to avoid early initiation of tasks, in order to prevent potential waste of resources, and is therefore less capable of absorbing delays.

A complete analysis of delays requires to run multiple scenarios, including different amount of delays in various sets of subsystems. However, we can account for the large amount of uncertainty in complex systems, which are common in the space domain. For this purpose, we applied a stochastic version of SDDA, where the TRL provided by the AI agents was used to evaluate different amount of uncertainty in the expected time required to develop each subsystems. This uncertainty is propagated along the network according to the SDDA model, and the result is a stochastic Gantt chart, where the schedule shows the uncertainty in the completion time of the subsystems development, and suggests appropriate times to begin the development of each subsystem. Figure 13 shows the result of this analysis. Due to the initial low TRL of some of the involved technologies, the schedule presents initially large uncertainty. However, the analysis can be run again at later times, based on information collected during the development process itself. The right side of figure 13 shows the schedule resulting from SDDA analysis performed after 6 years, with a consequent reduction in the uncertainty.

5. Conclusions and Future Work

This work illustrated an application of Artificial Intelligence to create a preliminary space architecture database and to retrieve useful sources of data used to feed a set of SoS tools. The tools of the Analytic Work Bench provide analysis of the impact of dependencies between systems in the operational and in the developmental domain. AI agents were trained and used to identify useful sources for the parameters and inputs of the models used by the tools of the AWB. Methodological advancements and results include:

- Implementation and Training of AI agents capable of interpreting natural language descriptions. The agents use a score rather than a simple binary evaluation of keywords and can therefore use concepts and give a more detail assessment of the relevance of literature sources to topics of interest.
- Refinement of the agents through evaluation against existing repositories and interaction with Subject Matter Experts to improve the definitions and boost specific keywords and concepts within the agents.
- Use of filters on the sources identified by each agent provided information about disruptions and failures of specific subsystems. This data have been integrated with information from the Subject Matter Experts to build a SODA model of the NASA Gateway Habitat.
- Use of information about the agents related to a specific subsystems in order to sort each subsystem into categories in NASA TABS. These categories provide current and necessary Technology Readiness Level and development time, which have been used to build an SDDA model of the NASA Gateway Habitat.
- SODA analysis indicates that Fire Safety, Primary Support Structure, MMOD Protection, Power Distribution, and Avionics are the most critical elements in the operational domain, that is the subsystems which cause the highest cascading impact on other subsystems when disrupted. High criticality suggests technologies that might require enhancement or insertion to guarantee their robustness.
- SDDA analysis indicates that Structure, MMOD Protection, Power Distribution, and ECLSS subsystems are critical in the developmental domain. This criticality means that delays in the development of these subsystems and their associated technologies can cause the highest delays in the development of the whole habitat. High criticality suggests tasks that might require more attention, whether that means earlier initiation or higher resource investment.

Since the proposed methodology is constantly evolving and being improved based on user needs, we identified promising future directions of research. First, since the modeling phase can be time-consuming and still involve a large number of sources, we advocate the implementation of a space architecture database and the use of large amount of autonomy and Machine Learning, to support the humans-in-the-loop during the modeling phase.

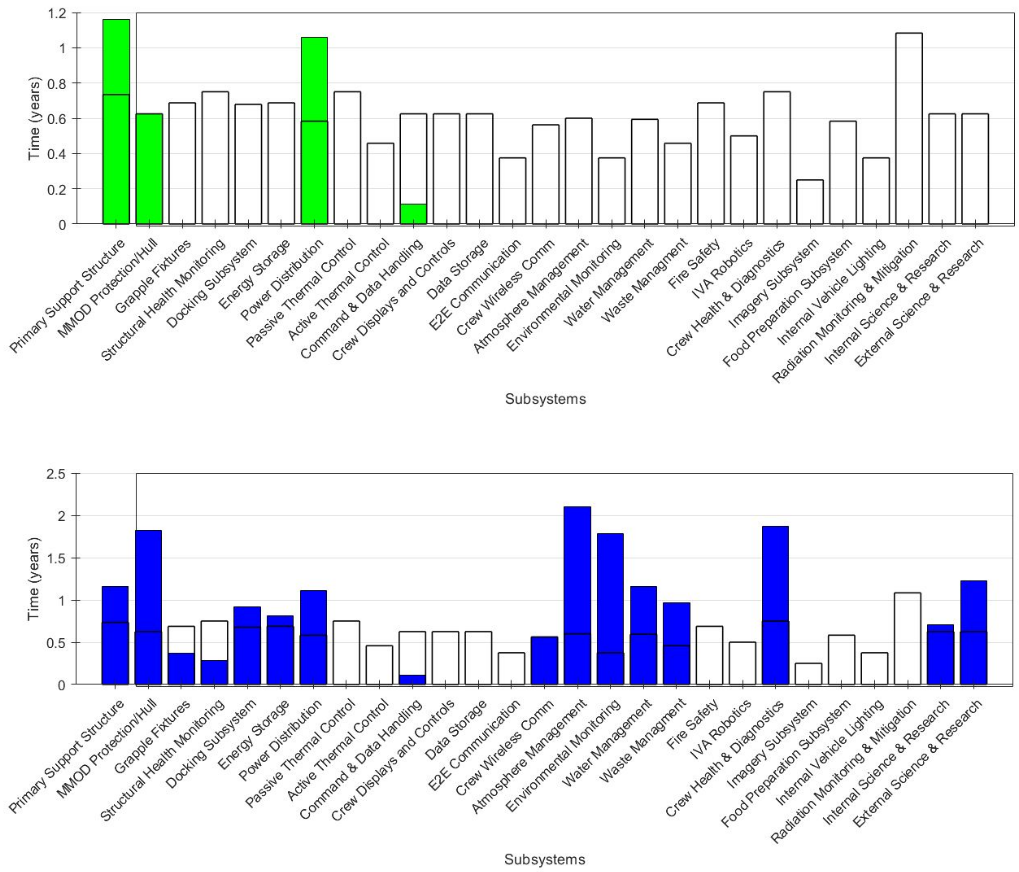


Figure 12. Initial and final schedule delays due to delays in individual subsystems. Top: SDDA model. Bottom: Conservative SDDA model. The Primary Support Structure and the Power Distribution are the most critical elements in the SDDA model. The structural elements, ECLSS, and Science Support subsystems have the highest unabsorbed delays in the conservative model. The conservative approach is less capable of absorbing delays, but it is also less prone to waste resources with too early initiation of tasks.

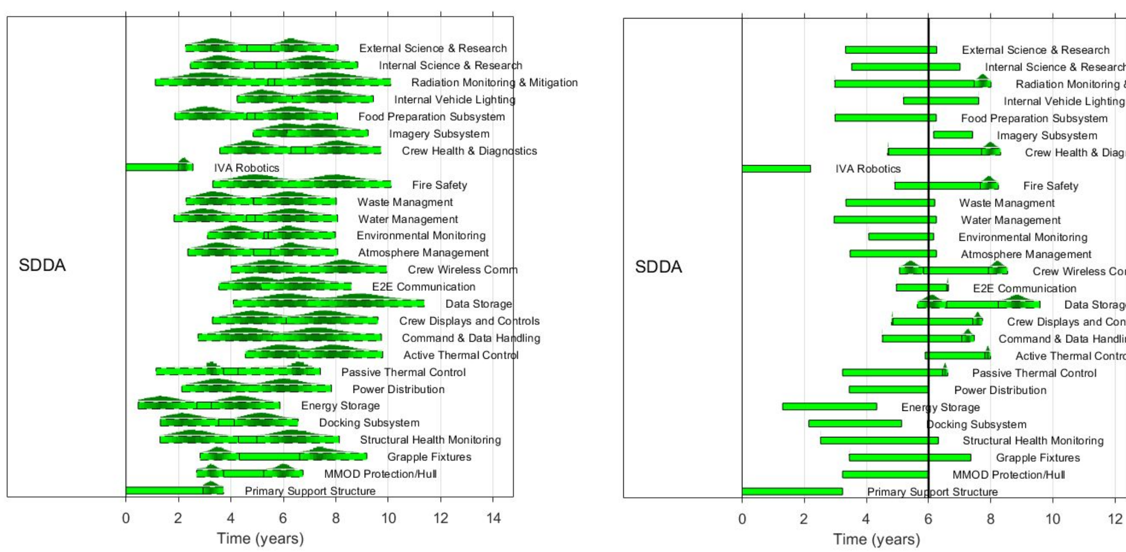


Figure 13. Stochastic SDDA analysis of the habitat development. Left: analysis at time t=0 shows large uncertainty in the schedule. Right: analysis at time t=6 years shows the reduced amount of uncertainty and a consequent shorter expected development time.

Effort is currently underway to apply this to missions being planned by other university and NASA teams, including the Gateway Habitat program. In addition to developing a set of agents for those subsystems, our roadmap includes expansion of the space architecture database to JANNAF, AIAA, space news sources and other credible resources needed by the SoS engineer. Furthermore, we propose the development of a web platform with the SODA and SDDA AWB tools integrated and available across NASA.

Finally, at the time of this publication NASA released a new structure for the technology roadmap, called the *NASA 2020 Technology Taxonomy*. Our goal is to train the agents against the new taxonomy, as well as to identify common traits with the objectives indicated in NASA's Decadal Survey, to bridge the gap between technological and scientific approach.

ACKNOWLEDGMENTS

This material is based upon work supported, in whole or in part, by the National Aeronautics and Space Administration (NASA) under Contract 80MSFC19M0004 with the Von Braun Center for Science and Innovation (VCSI) and Purdue University.

REFERENCES

[1] D. Selva, B. Cameron, and E. Crawley, "Patterns in System Architecture Decisions," *Systems Engineering*, vol. 19, no. 6, pp. 477–497, Nov. 2016.

[2] A. K. Raz, C. R. Kenley, and D. A. DeLaurentis, "System architecting and design space characterization," *Systems Engineering*, vol. 21, no. 3, pp. 227–242, May 2018.

[3] C. Guariniello and D. A. DeLaurentis, "Maintenance and recycling in space: functional dependency analysis of on-orbit servicing satellites team for modular spacecraft," in *AIAA SPACE 2013 Conference and Exposition*, 2013, p. 5327.

[4] S. Zusack, C. Guariniello, and D. DeLaurentis, "Operational dependency analysis of a human mars architecture based on the soda methodology," in *2018 IEEE Aerospace Conference*. IEEE, 2018, pp. 1–12.

[5] C. Guariniello, L. Mockus, A. K. Raz, and D. A. DeLaurentis, "Towards intelligent architecting of aerospace system-of-systems," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–11.

[6] C. Guariniello and D. DeLaurentis, "Communications, information, and cyber security in systems-of-systems: Assessing the impact of attacks through interdependency analysis," *Procedia Computer Science*, vol. 28, pp. 720–727, 2014.

[7] W. Zhang, Z. Li, W. Wang, and Q. Li, "System of systems safety analysis of gnss based on functional dependency network analysis," *International Journal of Applied Mathematics and Information Sciences*, vol. 10, no. 6, pp. 2227–2235, 2016.

[8] C. Guariniello, M. Grande, C. Brand, L. Durbin, M. Dai, A. Das-Stuart, R. Alexander, K. Howell, and D. DeLaurentis, "Quantifying the impact of systems interdependencies in space systems architectures," in *International Astronautical Congress*, 2019.

[9] M. W. Maier, "Architecting principles for systems-of-systems," in *INCOSE International Symposium*, vol. 6, no. 1. Wiley Online Library, 1996, pp. 565–573.

[10] A. P. Sage and C. D. Cuppan, "On the systems engineering and management of systems of systems and federations of systems," *Information knowledge systems management*, vol. 2, no. 4, pp. 325–345, 2001.

[11] J. Dahmann, G. Rebovich, J. Lane, R. Lowry, and K. Baldwin, "An implementers' view of systems engineering for systems of systems," in *2011 IEEE International Systems Conference*. IEEE, 2011, pp. 212–217.

[12] N. Davendralingam, D. DeLaurentis, Z. Fang, C. Guariniello, S. Y. Han, K. Marais, A. Mour, and P. Uday, "An analytic workbench perspective to evolution of system of systems architectures," *Procedia Computer Science*, vol. 28, pp. 702–710, 2014.

[13] Z. Fang and D. DeLaurentis, "Multi-stakeholder dynamic planning of system of systems development and evolution," *Procedia Computer Science*, vol. 44, pp. 95–104, 2015.

[14] C. Guariniello and D. DeLaurentis, "Dependency analysis of system-of-systems operational and development networks," *Procedia Computer Science*, vol. 16, pp. 265–274, 2013.

[15] N. Davendralingam and D. A. DeLaurentis, "A robust portfolio optimization approach to system of system architectures," *Systems Engineering*, vol. 18, no. 3, pp. 269–283, 2015.

[16] C. Guariniello and D. DeLaurentis, "Supporting design via the System Operational Dependency Analysis methodology," *Research in Engineering Design*, vol. 28, no. 1, pp. 53–69, Jan. 2017.

[17] P. R. Garvey, C. A. Pinto, and J. R. Santos, "Modelling and measuring the operability of interdependent systems and systems of systems: advances in methods and applications," *International Journal of System of Systems Engineering*, vol. 5, no. 1, pp. 1–24, 2014.

[18] C. A. Pinto and P. R. Garvey, *Advanced risk analysis in engineering enterprise systems*. CRC Press, 2016.

[19] NASA. (2015) Nasa technology roadmaps archive. [Online]. Available: www.nasa.gov/offices/oct/home/roadmaps/index.html

BIOGRAPHY

Cesare Guariniello *is a Research Scientist at Purdue's School of Aeronautics and Astronautics. He received his Ph.D. in Aeronautics and Astronautics from Purdue University in 2016 and his master's degrees in Astronautical Engineering and Computer and Automation Engineering from the University of Rome "La Sapienza". Dr. Guariniello works as part of the Center for Integrated Systems in Aerospace led by Dr. DeLaurentis, and is currently engaged in projects funded by NASA, the DoD Systems Engineering Research Center (SERC), Northrop Grumman, and Crane NWSC. His main research interests include modeling and analysis of complex systems and SoS architectures – with particular focus on space mission architectures – aerospace technologies, and robotics. Dr. Guariniello is a senior member of IEEE and AIAA.*

Thomas B. Marsh *is COO at ai-one inc., VP Analytics at ISC International and CEO at Bintel Inc. He led the Topic Mapper project at NASA MSFC and consulted on the RS-25 Engine Cost Reduction team. He is an expert in text analytics, Tableau data visualizations and knowledge management solutions. He served as Chief Operations Officer (COO) of organizations with primary responsibility for marketing, finance and product management. He led the development strategy and deployment support for text analytics, knowledge management and intelligent agent systems. He has international experience in startup, turnaround and growth situations in the medical device, marketing and software industries for both public and private companies, primarily B2B selling to large Fortune 500 firms. Previously, Tom was Chief Executive Officer (CEO) of a data management and marketing services business and before that Viking Systems, a publicly listed supplier of 3D vision systems for surgery, where he led several funding rounds and built the medical sales, marketing and clinical organization. He majored in Economics at Williams College..*

Ron Porter *is an independent consultant specializing in strategic assessment and tactical execution. Before becoming a consultant, he retired from a 38 year career at NASA. While at NASA, he served in many capacities including Senior Advisor to the NASA Associate Administrator and Senior Advisor to the NASA Administrator for Strategy and Plans. He also led the Marshall Space Flight Center Nuclear Systems Program Office, the Biotechnology Program Office, and served as the Chief of the Vehicle Integration Systems Engineering Division and Acting Space Launch System (SLS) Lead System Engineer. Mr. Porter holds a bachelor's degree in Mechanical Engineering from the University of Tennessee and has received both the Exception Achievement and Outstanding Leadership medals from NASA. He is an Associate Fellow of AIAA.*

Christopher Crumbly *is the Senior Director for Space and Defense Programs at the University of Tennessee Space Institute (UTSI). Prior to accepting the UTSI appointment he served as Executive Director of the technical nonprofit Von Braun Center for Science and Innovation (VCSI). After a 25-year career at NASA, Mr. Crumbly retired and entered the business community as the Vice President for Business Development, Civil and Commercial Space at Teledyne Brown Engineering. While at NASA, Mr. Crumbly managed the Space Launch System (SLS) Program's Spacecraft/Payload Integration and Evolution Office. In addition to serving in management positions at NASA Marshall, he also served as Special Assistant to NASA's Deputy Administrator and Senior Space Policy Analyst at the White House Office of Science and Technology Policy. Mr. Crumbly holds both bachelor and master's degrees in aerospace engineering from Auburn University and is a graduate of the Program for Management Development at the Harvard Business School. Mr. Crumbly serves on several professional and civic boards including the Huntsville Space Club, the American Astronautical Society, and the Greater Huntsville Section of the American Institute of Aeronautics and Astronautics (AIAA) where he serves as the Region II Deputy Director for Public Policy. He is an Associate Fellow of AIAA. .*

Daniel DeLaurentis *is a Professor at Purdue's School of Aeronautics and Astronautics. He joined the faculty in 2004. Dr. DeLaurentis is the inaugural Director for Purdue's Institute for Global Security and Defense Innovation. He is also Director of the Center for Integrated Systems in Aerospace (CISA), which is home to over 15 additional faculty and staff, and leads the System-of-Systems Laboratory which includes graduate and undergraduate students as well as professional research staff. His primary research interests are in the areas of problem formulation, modeling and system analysis methods for aerospace systems and SoS with particular focus on network analysis.*