IAC-18-D1.4A.6

System-of-Systems tools for the analysis of technological choices in space propulsion

Cesare Guariniello[a]*, William O'Neill[a], Ashwati Das-Stuart[a], Liam Durbin[a], Kathleen Howell[a], Reginald Alexander[b], Daniel DeLaurentis[a]

[a] *School of Aeronautics and Astronautics, Purdue University, 701 W. Stadium Ave, West Lafayette, IN 47907*
[b] *NASA Marshall Space Flight Center, Martin Rd SW, Huntsville, AL 35808*
* Corresponding Author

Abstract

Difficulties in space mission architecture design arise from many factors. Performance, cost, and risk constraints become less obvious due to complex interactions between the systems involved in the mission; decisions regarding long-term goals can heavily impact technological choices for short-term parts of the mission, while conversely decisions in the near future will impact the whole flexibility of long-term plans. Furthermore, the space community is broadening its borders, and space agencies from different countries are collaborating with industry and commercial partners towards large-scale endeavors. This paradigm shift is prompting the development of non-traditional approaches to the design of space missions. This paper reports the results of the first year of a continuing collaboration of the authors to develop and demonstrate System-of-System engineering methodologies for the deep analysis of dependencies and synthesis of robust architectures in exploration mission contexts. We present the procedure that we followed to develop and apply our methodology, obstacles found, steps taken to improve the methods based on the needs of experts and decision makers, required data for the analysis, and results produced by our holistic analysis. In particular, we focus on the analysis of technological choices for space propulsion for a generic cislunar mission, including both complex interactions between subsystems in different type of propulsion and availability of different providers. We identify critical systems and sets of systems based on cascading effects of performance degradation, assessment of the robustness of different designs in the operational domain, and simultaneous analysis of schedule dependencies between the constituent systems.

Keywords: System-of-Systems, Dependencies, Propulsion, Technology

Acronyms/Abbreviations

AMCM	Advanced Mission Cost Model
AWB	Analytic Work Bench
CDH	Command and Data Handling
DDTE	Design Development Test and Evaluation
DoD	Department of Defense
DSH	Deep Space Habitat
ISPS	In-Space Propulsion System
MBSE	Model-Based Systems Engineering
MCC	Mission Control Center
NTR	Nuclear Thermal Rocket
PDS	Propellant Distribution System
PPE	Power Propulsion Element
PPU	Power Processing Unit
RCS	Reaction Control System
RPO	Robust Portfolio Optimization
SDDA	Systems Developmental Dependency Analysis
SEP	Solar Electric Propulsion
SME	Subject Matter Expert(s)
SoS	System-of-Systems
SODA	Systems Operational Dependency Analysis

1. Introduction

The design and architecture of space missions, as well as the use of Systems Engineering to ensure accuracy for the technical development of space systems, are well-established methodologies [1, 2], which rely on more than half a century of expertise. To address the vast design space associated with space systems and space missions, government agencies and industry typically rely on technical teams with the needed expertise [3]. However, the traditional approach alone is no longer sufficient to work within the new, evolving, and complex context of space mission design. The size and complexity of current and future space missions are no longer characterized by a single entity of control, and the need arises for a systemic view across the set of stakeholders, variables and metrics, and systems involved. Chasing the goal of optimizing every system accounting for all individual stakeholder desires may be unreasonable. This systemic view, instead, can objectively analyze a large amount of diverse technological choices for space missions and compare different architectures to inform stakeholders of features and consequences of different technological choices.

Since space missions are now characterized by a large number of complex interdependent systems, often in an evolving scenario (including changes in policies and development of new technologies), a holistic view of entire space systems architectures is necessary to provide this systemic approach. This approach is not meant to replace the conventional approach to space mission architecture design and decision-making for technologies, but to be integrated with it, by adding considerations on the impact of dependencies between systems, on consequences of technological choices on long-term cost and performance, and on risk at the highest level of abstraction.

The problem of analyzing technological choices and their impact on the design of space missions as a whole is well suited to be treated as a System-of-Systems (SoS) problem. Maier [4] recognizes distinguishing characteristics of a SoS: (1) Operational independence of the individual systems within the SoS, (2) Managerial independence, in that the elements have unique operations and purposes based on their owner's intent and can be provided by different stakeholders, (3) The possibly changing behavior of the SoS as elements are removed or added from the network, (4) The emergent behavior where properties of the whole SoS due to the interactions of the elements differ from what would result from the elements considered individually. It is evident how the SoS traits are present in problems in the space domain. Various authors already identified systems and missions in the aerospace domain that can be treated as Systems-of-Systems [5], for example satellite formations [6] and in-orbit assembly [7]. Based on the considerations expressed above and on previous experience in the study of complex systems in other sectors, including Defense and Air Transportation, SoS engineering researchers at Purdue University advocated a widespread use of SoS methodologies for human space exploration [8]. The authors proposed the use of an Analytic Work Bench (AWB), a suite of tools and methods capable to provide the necessary top-level holistic assessment of complex architectures [9]. Some of the methods deal with uncertainty and risk in development schedule [10]. Other tools consider the operational aspects of architectures [11, 12], and some have been applied to aerospace problems [13].

In a continuing effort to develop the AWB and apply the methods to the analysis of problem in the space domain, the author initiated a collaboration with Subject Matter Experts (SME) at NASA. In particular, after an initial period of demonstration and validation of capabilities of the AWB, the methodology has been further tailored for problems of interest to space architectures decision makers. The tools proved themselves flexible enough to provide analysis of lunar architectures, based on the Gateway, after having initially been used for evaluation of Mars architectures. The current research task is focused on the evaluation of technological choices that result in alternative architectures for space exploration. The architectures are evaluated in terms of cost, performance, operational risk, possible cascading failures and criticalities, and schedule. In this paper, after briefly introducing the methods from the AWB applied in this study, we present preliminary results on the impact of different technological choices for space propulsion and launchers.

2. The SoS Analytic Work Bench and its evolution

The SoS Analytic Work Bench was initially developed to address the need, recognized by the US Department of Defense (DoD), for new methodologies and technical tools to manage the development of SoS architectures [14, 15]. This suite of tools addresses different aspects of complex architectures, including operability, cost, performance, schedule, and robustness. It relegates the management of complexity to the tools while leaving the decisional power with the user, whose tradeoff choices are supported by results and insights provided by the AWB.

Tools from the AWB have been used in different applications, for example Global Navigation Satellites Systems [16] and Cybersecurity [17]. Recently, a collaboration has been initiated with NASA, which is providing feedback for the improvement and further development of tools in the AWB, and part of the input data required by the tools. Preliminary results of the analysis of potential space mission architectures are providing NASA with useful insights into holistic features of the architectures, including criticalities and impact of technological choices. Three tools have been used to obtain the results described in this paper: Robust Portfolio Optimization (RPO), Systems Operational Dependency Analysis (SODA), and Systems Developmental Dependency Analysis (SDDA).

2.1 Robust Portfolio Optimization (RPO)

Robust Portfolio Optimization (RPO) is a methodology for comparing different selections, or portfolios, of systems that combine to meet System-of-Systems requirements and effectively accomplish an overall goal. These systems are governed by constraints that come from technological, operational or budgetary considerations as well as system to system integration. This method has its roots in financial engineering where it is used to maximize expected profit while minimizing the combined risk of a collection of investments. As a result, it is well suited for comparing risk and reward of selected options. It has since grown to apply to engineering problems by allowing constraints to be enforced both on the interaction between systems and the resulting portfolio of systems. In the engineering sense, this can be used to help mission architects choose which technologies to invest in given uncertain capabilities.

What differentiates RPO from other forms of multidisciplinary design optimization methods, is its basis in network theory. Each system is integrated within the larger SoS by its respective capabilities and requirements. These requirements are satisfied by other nodes in the network allowing for a collaborative operation. The interactions between nodes are modelled by the following five rules grounded in network theory: finite capability, requirements, compatibility, relay, and bandwidth.

The approach to solving a problem of this class is similar to the traditional systems engineering V-model. First, overall mission objectives and requirements are defined. Next, a library of available systems with values for each of the five network constraints are defined. In terms of space exploration architectures, these systems can range from different launch vehicles, habitat systems, power systems, propulsion systems and crew return vehicles (non-exhaustive list). Each system has a different associated cost, performance, schedule impacts and set of requirements necessary to function. This lends well to the current status of the space industry, where there are often several providers of systems with similar functionality. These systems combine to form a library of possible choices that are used in the optimization. A mixed integer optimization scheme is applied to find a portfolio of systems that maximizes key mission objectives given the network and integration constraints of the individual systems.

The portfolio optimization approach is made robust by the inclusion of uncertainty in the calculation of risk. Several risks ranging from operational, budget, and schedule can be associated with each available system and how that propagates through the architecture. Architectures can then be compared on the basis of cost, risk, and performance.

Key to the validity of results with this method is the accuracy of lifecycle cost and schedule components. For each potential space system used in this methodology, a cost estimation technique to assess the development, production and operation of each system in terms of cost and schedule was applied. Cost and schedule estimates were categorized into systems that are currently operational or available for purchase, those that are near term with published cost and schedule data and systems yet to be developed. In order to assess the development and production cost and schedule impacts of undeveloped systems, a modified version of NASA's Advanced Missions Cost Model (AMCM) was applied to the systems within the candidate system library [18].

To form an accurate architecture timeline, certain scheduling constraints were imposed on the architecture. Basic rules governing the beginning and end of the different product lifecycles, including development, production and operation phases, were applied as constraints across the architecture. Duration of these phases were either estimated through the use of AMCM, found in published literature or set to zero to represent currently available systems. The beginning and end times were then used as variables within the optimization such that schedule could be either optimized or constrained to specific architecture level requirements. This approach lends well to multi-objective optimization in which objectives ranging from total budget, different risks, and overall mission performance objectives can be compared.

2.2 Systems Operational Dependency Analysis (SODA) and Systems Developmental Dependency Analysis (SDDA)

The Dependency Analysis Methodology, based on the concepts of Functional Dependency Network Analysis [19, 20], assesses the effect of dependencies among systems in a SoS, both in the operational (SODA) and in the developmental (SDDA) domain.

Both methods are based on a parametric model of the behavior of the system and on a network representation of systems architectures, where the nodes represent the constituent systems and the capabilities that the SoS has to achieve. The edges represent the operational or developmental dependencies, as shown in Fig. 1. Low-level (systems-level) SODA network for chemical propulsion systemsFig. 1.

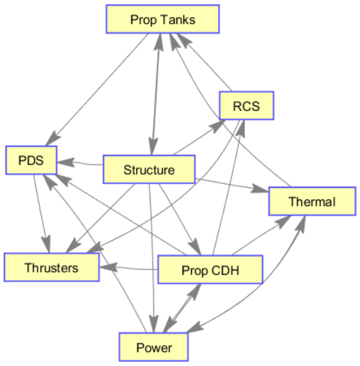

Fig. 1. Low-level (systems-level) SODA network for chemical propulsion systems

These dependencies are modeled with a small number of parameters, which quantify the impact of the dependencies on the behavior of the whole SoS. The representation of an SoS as a network prevents the methods from being domain-dependent and allows for their application across various classes of problems. This approach makes the model more intuitive, with parameters directly related to features of the dependency. SODA can then model the impact of cascading failure, offering a quantitative alternative to risk matrices, which also includes partial disruptions and multiple paths of propagation. SDDA can model the impact of delays on development and production schedule, accounting for partial overlapping of systems development. Both methods can then support informed decision making in design and update of systems and SoS architecture, reducing the amount of interrogative operations, such as simulation, required to obtain the necessary information.

The parameters of the model can be quantitatively linked to a range of possible input sources, including (but not limited to) experiments, historical data, and subject matter expert evaluation. The combined effort of researchers at Purdue and NASA is providing better modeling of case studies. Using Dependency Analysis methodology, designers and decision makers can quickly analyze and explore the behavior of complex systems in the operational and developmental domain and evaluate different architectures under various working conditions and policies. The architectures can be provided by RPO or by the user. Various metrics of interest (for example robustness, criticalities, and delay absorption) can then be added to the metrics assessed by RPO, in order to explore and support tradeoff in multi-dimensional trade space in early phases of the design process. The final outcome is a quantitative and objective support to the process of technical and technological decision-making behavior, concept selection, risk prevention, and development schedule.

3. Application of the Analytic Work Bench in support of strategic decisions in space mission design

The process of improving the AWB and tailoring it to the need of different users resulted in the development of formal procedures for the application of tools and methodologies in the AWB for specific problems. In the current effort, of which preliminary results are reported in this paper, the AWB has been applied in support of strategic decisions in the design of space mission architectures. First of all, since the tools in the AWB are domain independent, the research focus could easily switch from analysis of mission for direct exploration of Mars to analysis of architectures that include systems for operation in Lunar orbit and on the surface of the Moon. Second, the support provided by SME and the information gathered from literature review were used for a sequential process, where a library of available choices for systems assigned to perform the required functions is used to generate potential architecture designs with RPO. The architectures are generated based on considerations of cost and analyzed in terms of metrics of performance and schedule. The best architectures are analyzed at multiple hierarchical levels (entire architectures, which constitute a SoS, and some of their component systems expanded in networks of subsystems with their dependencies) with SODA and SDDA to provide further insight into holistic properties of the architectures. The metrics generated by analysis with the various tools are used for educated decision on final design, accounting for systems properties, holistic SoS properties and cost, performance, schedule, and risk associated with each architecture. A diagram of the process is shown in Fig. 2.

4. Case study: propulsion systems and launchers

This combined effort of System-of-Systems researchers with SME from NASA has a broad objective of identifying key technologies and design choices that will result in the *best* options for space mission architectures, based on the considerations exposed in the previous sections. To demonstrate the procedure and the analysis that is being conducted, we show some of the results pertinent to choices of in-space propulsions systems and type of launchers.

4.1 RPO analysis

An enhanced version of the robust optimization method was applied to a space exploration architecture. The methods previously described are applied to a generic cislunar space exploration architecture scenario consisting of a Deep Space Habitat (DSH) orbiting the moon and robotic lunar landers deployed to the lunar surface. This specific mission is examined at a high level in terms of cost, performance and robustness. The Candidate System Library for this study, includes several options for many of the systems required for a cislunar mission, but is not all inclusive. For demonstration purposes, there are many specific choices for launch vehicles and in-space propulsion systems, however the list is non-exhaustive especially in relation to vehicle sizing. The goal of the research effort is to demonstrate how this methodology can be useful in making architecture-level decisions, and support tradeoffs, rather than to dictate exactly how the space architecture should be designed. Preliminary results provide some interesting findings, and those related to propulsion and launcher choices are presented here.

As previously mentioned, one of the strengths of RPO is the ability to investigate multiple sources of architecture value. In this vein, a mission architect can trade off two or more individual objectives through the use of a weighted multi-objective function used within the optimization scheme.

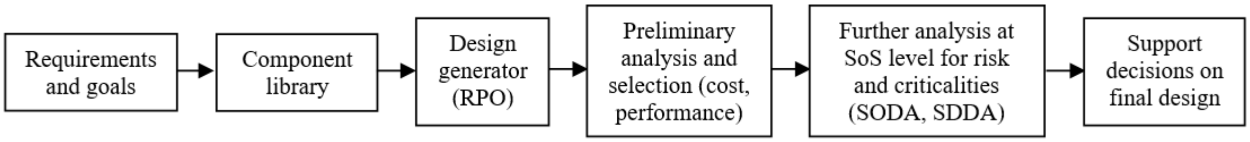

Fig. 2. The AWB process for analysis of space mission architectures

Fig. 3 shows the result of applying the RPO methodology with a suitable weighted objective function, a pareto frontier of the tradeoffs between a measure of architecture utility and total architecture cost. For different levels of total architecture cost, different systems are selected by the optimization scheme that obey the imposed network constraints in terms of capabilities, requirements and compatibilities. For weightings in which utility is valued more than total architecture cost, larger and more advanced DSH modules are required, which require larger launch vehicles and in-space propulsion systems which have unique associated lifecycle costs. The allocation of systems can be seen in Table 1 for the various weightings of the objective function.

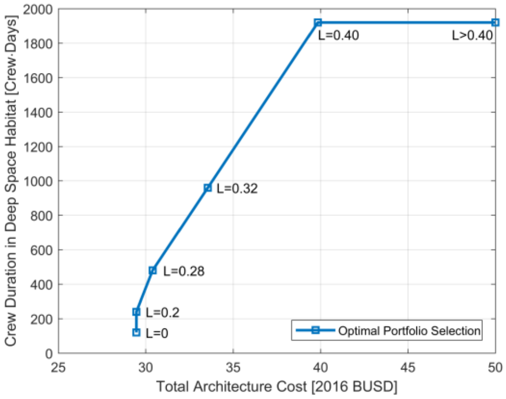

Fig. 3. Pareto frontier of total architecture cost versus a measure of architecture utility (total occupation of a cislunar deep space habitat).

Table 1. System allocation for pareto frontier of total architecture cost versus a measure of architecture utility

	L (Objective function weighting factor)	0	0.2	0.28	0.32	0.4+
	Total Crew Duration [Days]	120	240	480	960	1920
	Total Architecture Cost [2016 MUSD]	29317	29467	30397	33550	39844
System Allocation	4 Person Crew, 30 Day	1	0	0	0	0
	4 Person Crew, 60 Day	0	1	0	0	0
	4 Person Crew, 90 Day	0	0	0	0	0
	4 Person Crew, 120 Day	0	0	1	2	4
	Deep Space Habitat 1	1	1	1	1	1
	Deep Space Habitat 2	0	0	0	0	0
	Deep Space Habitat 3	0	0	0	0	0
	Moon Cargo Lander & Systems	0	0	0	0	0
	Orion	1	1	1	2	4
	Commercial Heavy Lift	0	1	1	2	4
	Commercial Super Heavy Lift	4	3	3	4	6
	Commercial Medium Lift	0	0	0	0	0
	Government Super Heavy Lift 1B	0	0	0	0	0
	Government Super Heavy Lift 1A	0	0	0	0	0
	Logistics Module 1	0	0	1	0	0
	Logistics Module 2	0	0	0	2	0
	Logistics Module 3	0	0	0	0	0
	Logistics Module 4	0	0	0	0	0
	PPE	1	1	1	1	1
	Science Airlock	1	1	1	1	1
	Crew Airlock	1	1	1	1	1
	Robotic Arm	1	1	1	1	1
	Prop Storage	1	1	1	1	1
	Lunar Orbit Science Payload	1	1	1	1	1
	SEP In-space Propulsion System	0	0	0	0	0
	NTR In-Space Propulsion System	0	0	0	0	0
	Chemical In-Space Propulsion System	3	3	3	4	6
	Landers	3	3	3	3	3
	Mission Control Center	1	1	1	1	1

One of the recent additions to the RPO methodology is the inclusion of scheduling within the optimization scheme by accounting for the development and production of each allocated system and how that affects the time when systems become operational. The impact of how these constraints are implemented can be seen in Fig. 5, in which each systems development and production is constrained by a set of scheduling rules. Demonstrated in Fig. 4 is a tradeoff comparison of a multi objective optimization of the year of first crewed launch and total architecture cost with the associated system allocation seen in Table 2. The impact of how an architect values the ability to fly the first crewed mission and how that affects the cost of the entire architecture can be clearly seen. Certain systems may be advantageous to flying a crewed mission sooner but have negative impacts to the resulting total architecture cost even after the first mission has been flown. This is primarily due to the large Design Development Test and Evaluation (DDT&E) cost of space systems.

Additional metrics can be examined through the RPO methodology. Metrics such as operational robustness, financial robustness, annual budget impacts, and other metrics of architecture value are demonstrated in [21, 22].

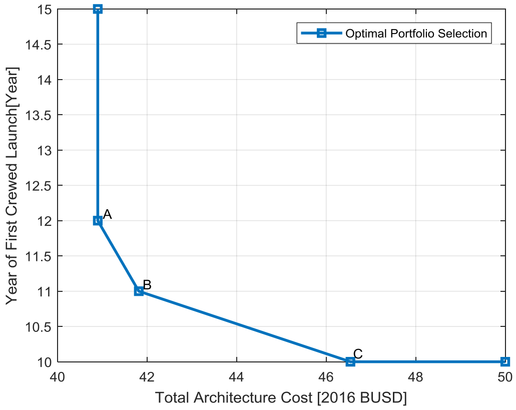

Fig. 4. Pareto frontier of total architecture cost versus year of first crewed flight to a Cislunar Deep Space Habitat

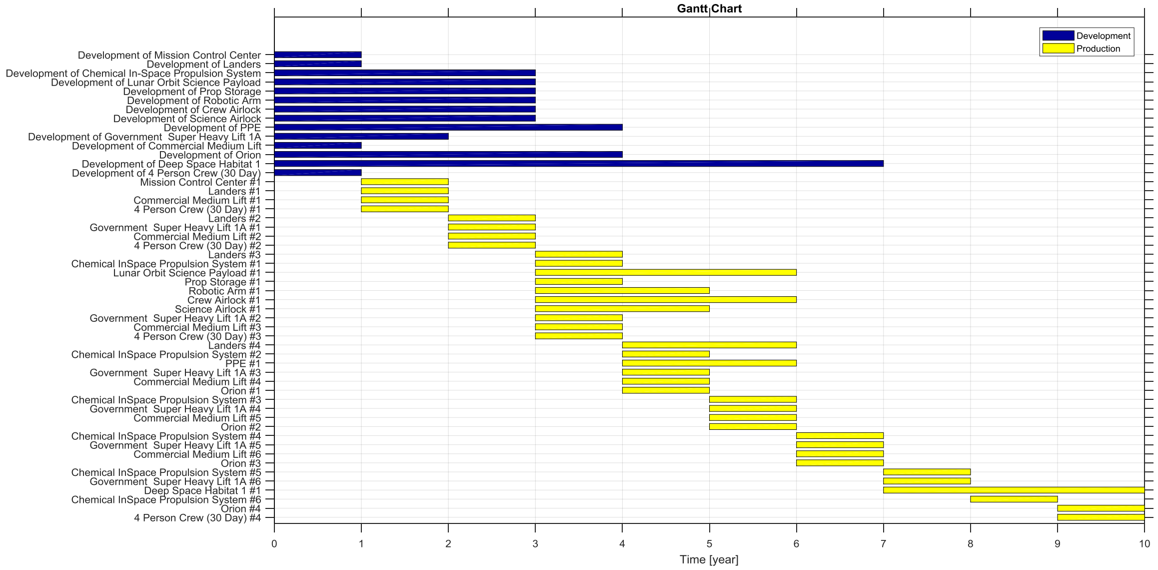

Fig. 5. Scheduling of system development and production cycles and the impact of system selection

Table 2. System allocation of pareto frontier of total architecture cost versus architecture readiness time

	Portfolio	A	B	C
	Year of First Crewed Flight [years]	12	11	10
	Total Architecture Cost [2016 MUSD]	40898	41814	46392
System Allocation	4 Person Crew, 30 Day	4	4	4
	Deep Space Habitat 1	1	1	1
	Deep Space Habitat 2	0	0	0
	Deep Space Habitat 3	0	0	0
	Moon Cargo Lander & Systems	0	0	0
	Orion	4	4	4
	Commercial Heavy Lift	2	4	7
	Commercial Super Heavy Lift	5	4	0
	Commercial Medium Lift	3	3	1
	Government Super Heavy Lift 1B	0	0	0
	Government Super Heavy Lift 1A	0	0	4
	Logistics Module 1	0	0	0
	Logistics Module 2	0	0	0
	Logistics Module 3	0	0	0
	Logistics Module 4	0	0	0
	PPE	1	1	1
	Science Airlock	1	1	1
	Crew Airlock	1	1	1
	Robotic Arm	1	1	1
	Prop Storage	1	1	1
	Lunar Orbit Science Payload	1	1	1
	SEP In-space Propulsion System	0	2	3
	NTR In-Space Propulsion System	5	4	0
	Chemical In-Space Propulsion System	0	0	4
	Landers	4	4	4
	Mission Control Center	1	1	1
	Landers	3	3	3
	Mission Control Center	1	1	1

4.2 SODA analysis

Based on systems selection provided by RPO optimizations process, we ran SODA analysis on the optimal architecture for minimum time (which uses commercial heavy lifters and super heavy lifters, and chemical in-space propulsion) and on a more performing but more expensive and longer-term architecture that uses Nuclear Thermal Rockets (NTR) for in-space propulsion.

Fig. 6 shows an example of high-level SODA network, including an entire architecture of systems for exploration of Lunar space, and their operational dependencies. The choice of systems comes from the optimization run with RPO, while the dependencies have been modeled based on information from the SME and literature review. Some of the nodes in the holistic network have been expanded into lower-level dependency network, like the one shown in Fig. 1. Since this paper is focusing on choices for propulsion systems, we modeled low-level SODA networks for chemical propulsion systems, Nuclear Thermal Rockets (NTR), and Solar Electric Propulsion (SEP).

Since SODA does not use a simple binary model for failure impact and propagation, but models partial disruptions and non-linear operational dependencies, information and data cannot be found directly in existing literature. We modeled the expected operability of the systems based on expertise and available information. Operability is associated with a normalized measure of performance of the system. More details about the concept of operability and the SODA model are described in [12].

4.2.1 low-level

For each of the three options for in-space propulsion (chemical, NTR, and SEP), we modeled the network of required subsystems and their interactions. Then we ran analysis in two phases: in the first, we assumed a probability distribution of the internal status of each of the subsystems and calculated the distribution of the operability of each of the systems, which depends on the internal status and on the inputs received by the feeder systems. Disruption of a system can be due to internal failures, modeled as a lower value of internal status, or to propagation of failures from other systems, which happens according to the SODA model of dependencies. The expected value of operability is associated with the likelihood of the system working at an adequate level of performance, and this first part of the analysis provides an improvement on qualitative likelihood levels of risk matrices, by adding a systemic view and a quantitative model which accounts for interactions. In the second phase, SODA evaluates the impact of different amount of disruptions in different systems and subsystems. This part of the analysis is used to identify which subsystems and systems are most critical to the behavior and performance of the entire network. In this phase, SODA is adding quantitative dependency-based assessments to risk impact evaluation.

Fig. 7 shows the results of the first phase for the three propulsion systems. All three systems exhibit similar expected value of the final node, with a value consistent with historical rates of propulsion systems failures.

Fig. 8 shows the results of the second phase of analysis for the three propulsion systems. The gradual disruption of each subsystem is simulated (Self-Effectiveness, which is a measure of the internal status, decreasing from 100 to 0) and the impact of the disruptions on the final node of the network is coded in color, with green indicating nominal operability, yellow indicating sub-nominal, and red indicating critical status. The three propulsion systems show very different behavior: the most critical subsystems for chemical propulsion are the Reaction Control System, the Structure, and the Propellant Distribution System. NTR shows similar results, but the most critical subsystem is the Thermal Control System. For SEP, Power and the Power Processing Unit are highly critical. Table 3 summarizes the outcome of low-level SODA analysis of propulsion systems. Robustness ranges between 0 and 1 and quantifies the general capability of the network to withstand disruptions.

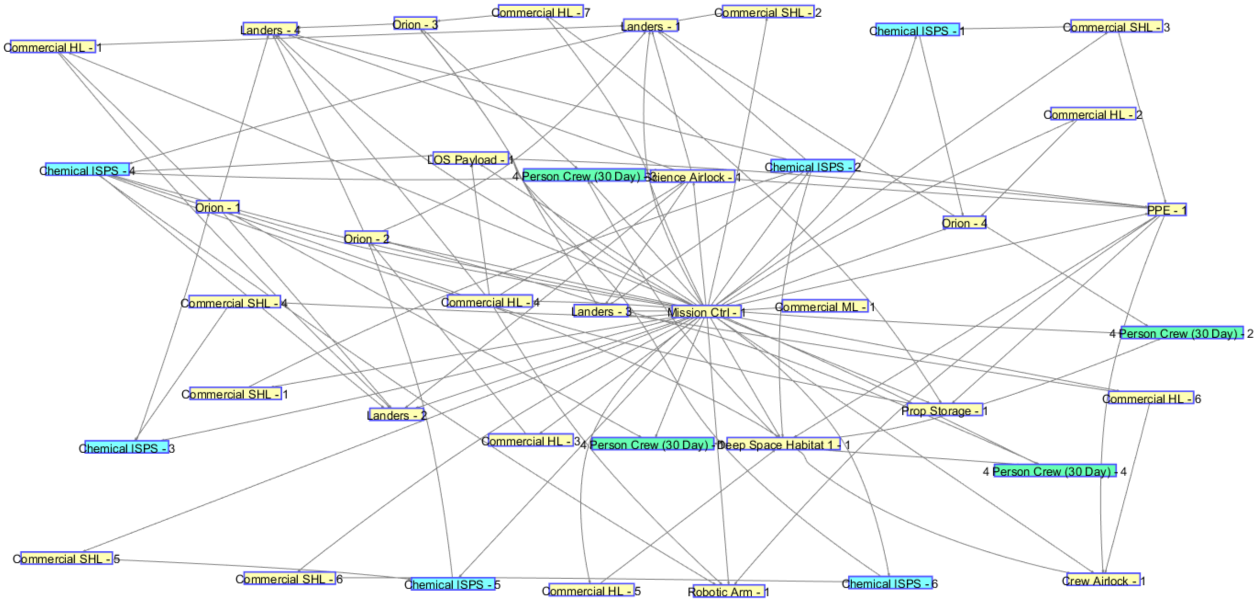

Fig. 6. High-level architecture with chemical in-space propulsion. Green nodes are objectives, cyan nodes are nodes that have been expanded into lower level networks

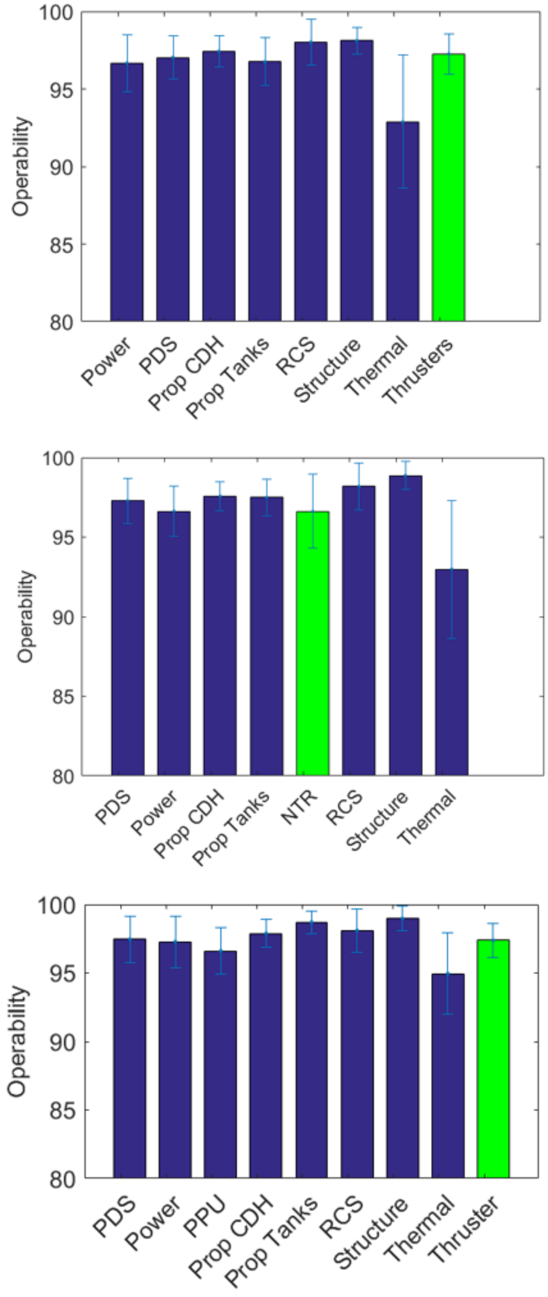

Fig. 7. Expected Value of operability of subsystems in propulsion systems. Top: chemical. Center: NTR. Bottom: SEP. Green bar is the end node. Error bars indicate 1σ standard deviation

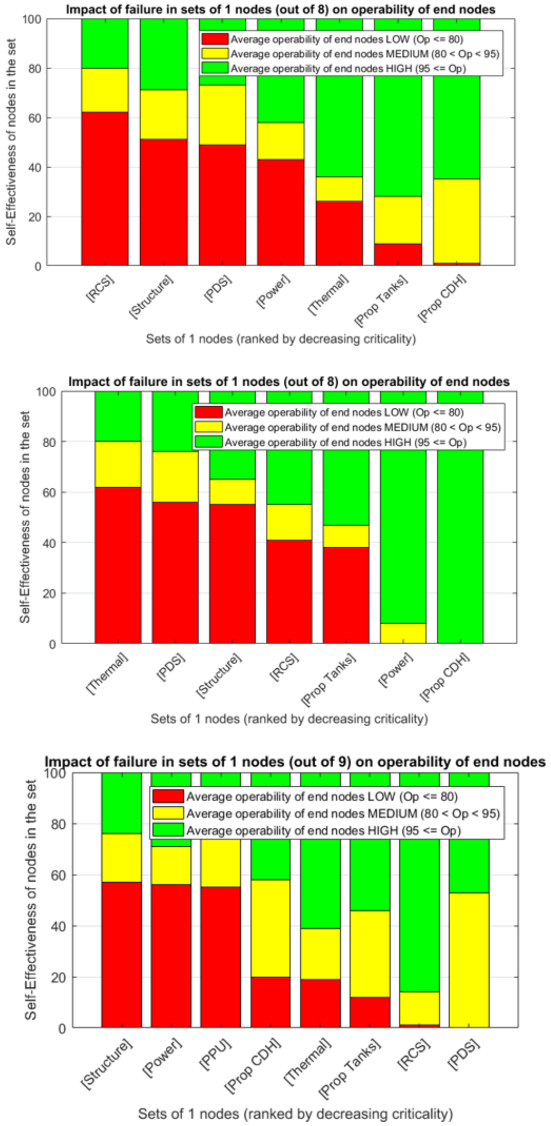

Fig. 8. Impact of disruptions in the propulsion subsystems. Colors indicate the nominal, sub-nominal, and critical status of the end nodes when the system indicated at the bottom of the bar experiences increasing disruptions. Top: chemical. Center: NTR. Bottom: SEP

Table 3. Summary of outcome of low-level SODA analysis of propulsion systems

	Chemical	NTR	SEP
E(Op) of end node	97.2	96.6	97.4
Robustness	0.67	0.62	0.70
Most critical subsystems	RCS, Struct, PDS	Thermal, PDS, Struct	Struct, Power, PPU

4.2.2 high-level

Results from the low-level analysis provide model of the internal status of some of the systems analyzed in high-level architectural networks generated by RPO. The operational dependencies in these architectures have then been modeled according to the SODA approach (Fig. 6 shows a network of these dependencies). Analysis similar to the one performed at the low level is then used to analyze operational risks and propagation of disruptions and to assess criticalities and robustness of various architectures. These metrics add to the metrics generated by RPO to provide a full holistic perspective of the *goodness* of architectures.

Fig. 9 shows the results of the first phase of SODA analysis for a minimum-time-of-development architecture, which uses in-space chemical propulsion, and an NTR-based architecture. It can be noted how different technological choices impact the entire architecture: optimization with RPO resulted in a larger number of launchers for the chemical-based architecture, with more Heavy Lift and Super Heavy Lift launchers involved, and the need for six chemical-based in-space systems. The NTR-based architecture makes use of more Medium Lift launchers. In both

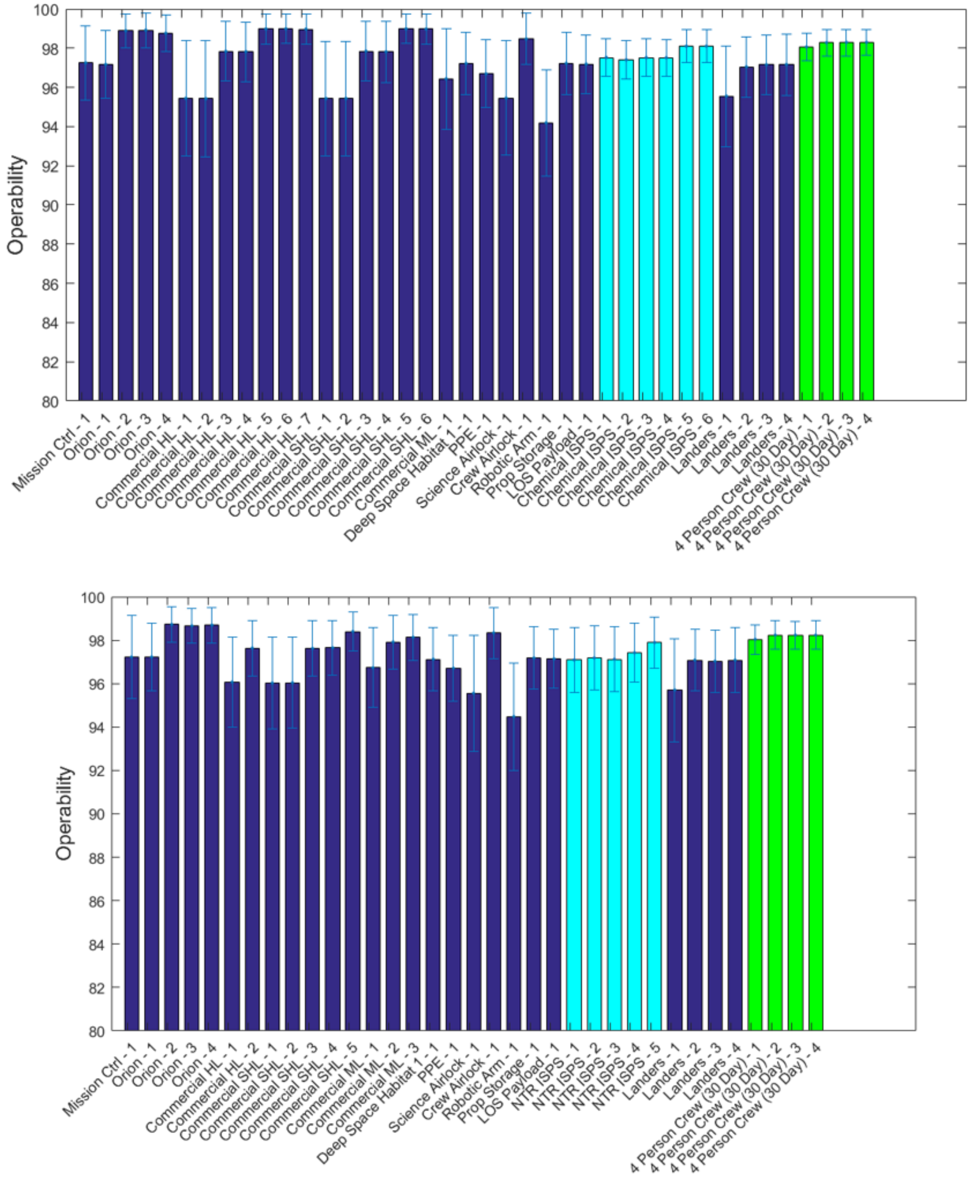

Fig. 9. Expected operability of systems in lunar architectures. Top: in-space chemical propulsion. Bottom: in-space NTR. Green bars are the end nodes. Cyan bars are nodes that are also modelled with a low-level subsystem network.

architectures, the first Heavy Lift and Super Heavy Lift launchers exhibit the lowest expected operability, as well as the first lander.

Fig. 10 shows the results of the second phase of SODA analysis. While the Deep Space Habitat and the Orion spacecraft are the most critical systems in both the chemical-based and the NTR-based architectures, a few differences can be noted. In general, disruptions of systems in the NTR-based architecture bring the whole mission down from sub-nominal to critical values more rapidly than in the chemical-based architecture (which has more systems and a better-known technology). In addition, the NTR in-space propulsion systems appear among the most critical systems.

Metrics based on SODA analysis are reported in Table 4, together with results from RPO and SDDA.

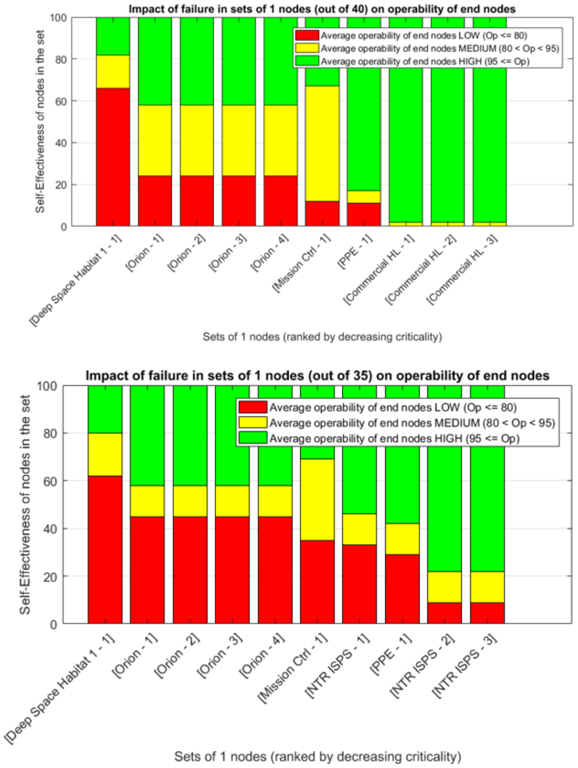

Fig. 10. Analysis of impact of disruptions in individual system on the whole architecture. Colors indicate the nominal, sub-nominal, and critical status of the end nodes when the system indicated at the bottom of the bar experiences increasing disruptions. Top: chemical. Bottom: NTR.

4.3 SDDA analysis

While considerations on schedule have recently been added to RPO, the SoS AWB has a tool which specifically addresses developmental dependencies and concerns about schedule, delays, and propagation of delays. Systems Developmental Dependency Analysis has been used to perform three different kind of analysis on the expected schedule for various lunar architectures. The first is a simple evaluation of the expected schedule, based on a model of the developmental dependencies that allows the user to keep into account partial dependencies (differently from PERT/CPM techniques), resulting in partial overlapping of development and production tasks. The second type of analysis, not shown in this paper, is a stochastic analysis. Based on uncertainty in development and production times, on Technology Readiness Levels, and on reliability of the various stakeholder, SDDA computes a probability distribution of the expected completion time of each of the tasks. This type of analysis is very useful to determine the risk associated with delays, and to assess the likelihood to meet deadlines, for example launch windows. The third type of analysis addresses delays and their propagation. Due to the partial overlap of development and production of the various systems, some of the delays can be partially or completely absorbed, while other delays will have a large impact on the completion of the whole architecture. Therefore, this analysis identifies the most critical systems in terms of development.

Fig. 11 shows the results of the basic deterministic analysis of the development of technologies and production of required systems for the optimal chemical-based architecture and the optimal NTR-based architecture. In this case, only developmental constraints are accounted for, resulting in a Gantt chart of the shortest development and production times. However, certain systems and also operationally dependent on other systems, and therefore cannot be in operation until the systems on which they depend are operational. Notwithstanding the partial overlap of systems development and production, the deterministic analysis confirms the findings of RPO that architectures that use NTR will take longer to be fully deployed. The development of NTR technology, rather than the actual production of the systems, is the major cause of this longer schedule.

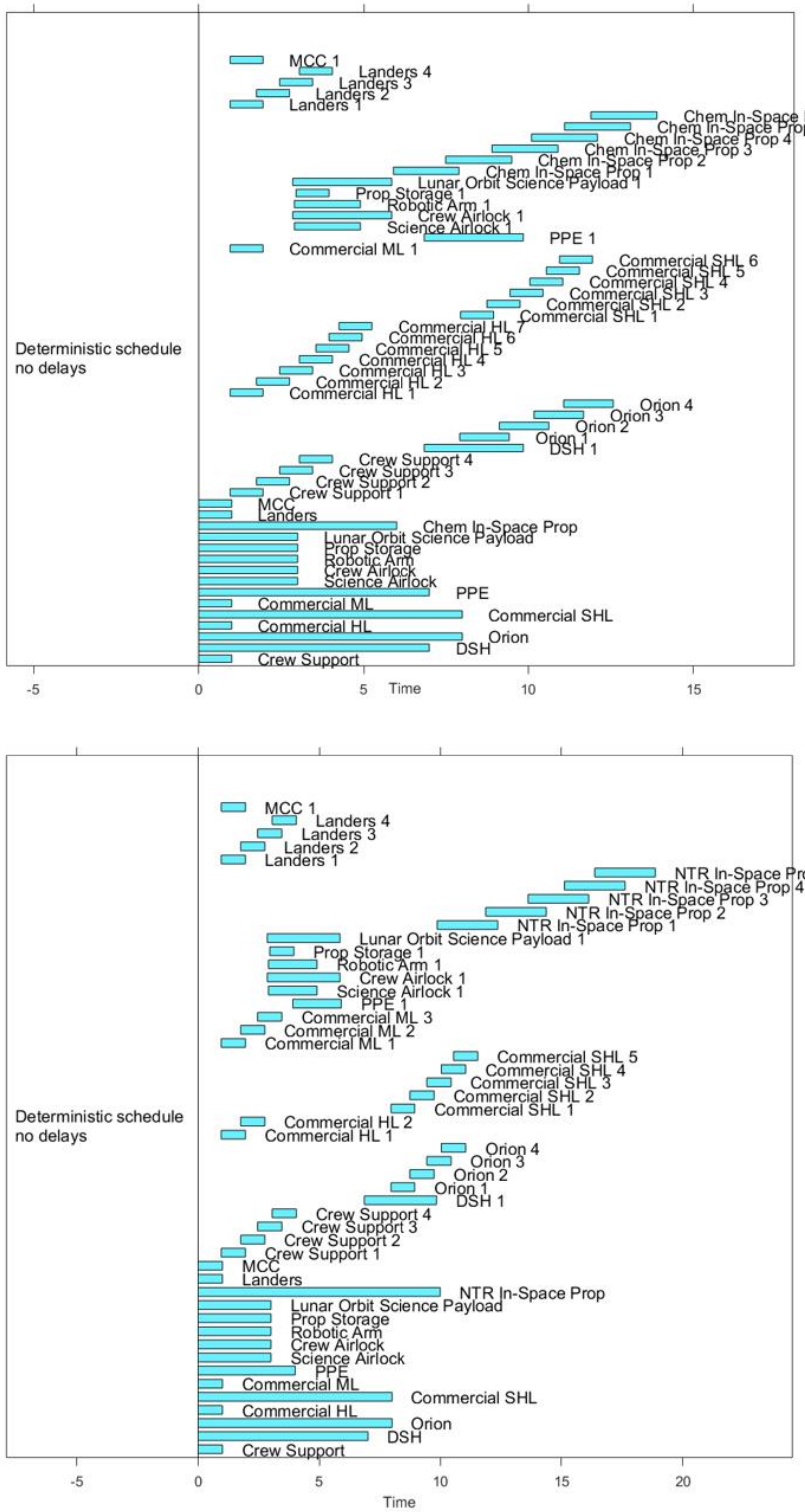

Fig. 11. Gantt chart of the development of technologies and production of systems in lunar architectures. Top: chemical-based. Bottom: NTR-based

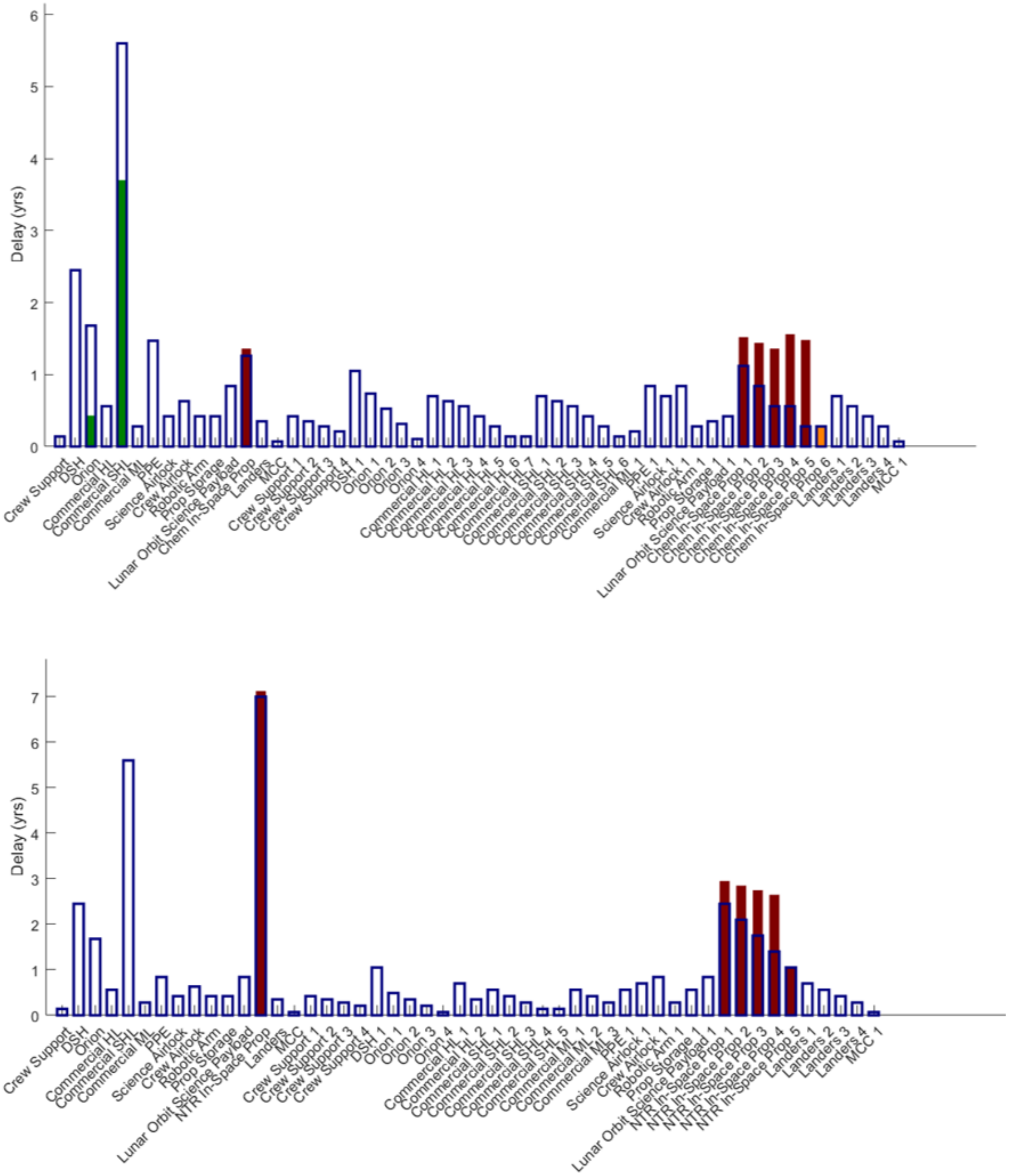

Fig. 12. Delays in individual systems (bar frames) and final impact on development of the whole architecture (filled bars). Top: chemical-based. Bottom: NTR-based

Fig. 12 shows the results of the analysis of delays and their impact in the two architectures. The blue bar frames indicate the initial delay in the development of the individual technology or system indicated at the bottom. If the rectangle is empty, the delay caused by that system will be completely absorbed and the whole architecture development will not be delayed. If the rectangle is partially filled with a green bar, the initial delay has been only partially absorbed and the height of the bar indicates the final delay in the development of the whole architecture. If the initial delay is not absorbed, or it causes an even greater delay on the whole architecture, an orange or red bar respectively indicates the final delay in the development of the whole architecture. In both architectures, the production of the in-space propulsion systems is among the most critical systems for what concerns impact of delays. However, it can also be noted that also delays in the development of Orion spacecraft and a commercial Super Heavy Lifter can have some major impact on the schedule of the whole architectures. Instead, the long time required for the development of NTR technology and the production of NTR-based in-space propulsion systems shown in Fig. 11 is so prominent with respect to the rest of the schedule that all other delays than those related to NTR can be fully absorbed without causing delays in the development of the whole architecture. It must be noted, however, that these results do not account for availability of resources, which might cause delays in other systems than the one directly affected by delays, due to the lack of available resources which would instead be available if the original schedule had been kept.

4.3 Outcome of AWB analysis

A summary of the outcome of the part of architectural analysis relative to propulsion and launchers is shown in Table 4, which combines results of RPO optimization, of SODA operational analysis, and of SDDA analysis of schedule. The metrics here shown are only a subset of the metrics assessed by this procedure, and the optimal architectures have been generated based on a limited amount of choices, but they demonstrate both the capabilities of the approach and the process we used.

Table 4. Metrics generated by RPO, SODA, and SDDA analysis for alternative optimal architectures

	Optimal chemical-based	Optimal NTR-based
Cost	$46.4B	$40.9B
Earliest first crewed mission	10 years	12 years
Number of launches	14	10
E(Op) of end nodes	98.24	98.11
Robustness	0.84	0.80
Most critical systems (operational)	DSH, Orion 1 to 4, MCC, PPE	DSH, Orion 1 to 4, MCC, ISPS 1 to 3, PPE
Expected completion time	13 years and 10 months	18 years and 9 months
Most critical subsystems (delays)	Chem development, Chem In-Space 1 to 5, Super Heavy Launcher, Orion	NTR development, NTR In-Space 1 to 4

5. Conclusions

We reported preliminary results of an ongoing combined effort of SoS researchers at Purdue university and SME at NASA. The research has the goal of improving and tailoring capabilities of a SoS Analytic Work Bench for application to the study of space mission architectures. At the same time, the study aims at providing support for technological choices for space exploration, offering a systemic, holistic view. This view accounts for multiple perspective involved with the large and multidimensional trade space which characterizes this family of problems, and offers useful insights that consider dependencies between system, constraints of budget and requirements of performance, and use models developed to address the complexity of SoS networks.

This paper presented only a small subset of the results that this research is producing. However, these results are a good example of how the proposed procedure and the use of SoS methodology in the AWB expands over the current approach and provides valuable models and a holistic overview of multiple metrics of interest in support of tradeoffs. The results shown in this paper are based on different choices of in-space propulsion systems and launchers. The optimal chemical-based solution is more expensive than the optimal NTR-based solution, but it has a shorter development schedule, allowing for an expected first crewed mission in 10 years. Robustness and expected operability are similar for the two architectures. The NTR-based architecture has more criticalities in the operational domain, while in the developmental domain it can absorb most of the delays in development and production of systems.

The proposed methodology is being constantly improved and exhibits large potential for even more capabilities. New methods can be added to the AWB, which is also being developed in the direction of Model-Based Systems Engineering (MBSE) and Artificial Intelligence. The use of MBSE will improve the modeling phase for the AWB tools, which is often made slow by the lack of readily available appropriate data to be fed into such novel tools. At the same time, MBSE will facilitate the introduction of this approach. Artificial Intelligence will instead be used to automatize parts of the process of modeling and analysis complex SoS networks, for example space mission architectures, with the AWB.

Acknowledgements

This material is based upon work supported, in whole or in part, by the National Aeronautics and Space Administration (NASA) under Contract 80MSFC18C0037, Purdue Space Technology Assessment and Planning.

References

[1] J. Wertz, D. Everett, J. Puschell, Space Mission Engineering: the new SMAD. Microcosm Press, 2011.

[2] P. Fortescue, G. Swinerd, J. Stark, editors, Spacecraft Systems Engineering, fourth ed., John Wiley & Sons, 2011.

[3] B. Blanchard, W. Fabrycky, Systems Engineering and Analysis, fifth ed., Prentice Hall International Series in Industrial & Systems Engineering, 2014.

[4] M. Maier, Architecting Principles for Systems-of-Systems, Sys. Eng., 1 (1998), 267–284.

[5] M. Jamshidi, ed., System of Systems Engineering, Innovations for the 21st Century, John Wiley & Sons, 2009.

[6] O. Walker, M. Tummala, J. McEachen, A System of Systems Study of Space-Based Networks Utilizing Picosatellite Formations, International Conference on System of Systems Engineering, 2010.

[7] T. Huntsberger, A. Stroupe, B. Kennedy, System of Systems for Space Construction, IEEE International Conference on Systems, Man and Cybernetics, 2005.

[8] C. Guariniello, W. O’Neill, T. Ukai, D. Dumbacher, B. Caldwell, D. DeLaurentis, Understanding Human Space Exploration, 67th International Astronautical Congress, Guadalajara, Mexico, 2016.

[9] N. Davendralingam, D. DeLaurentis, Z. Fang, C. Guariniello, S. Y. Han, K. Marais, A. Mour, P. Uday, An Analytic Workbench Perspective to Evolution of System of Systems Architectures, Procedia Computer Science, 28 (2014), 702–710.

[10] C. Guariniello, D. DeLaurentis, Dependency Analysis of System-of-Systems Operational and Development Networks, Procedia Computer Science, 16 (2013), 265–274.

[11] N. Davendralingam, D. DeLaurentis, A Robust Portfolio Optimization Approach to System of System Architectures, Sys. Eng., 18 (2015), 269–283.

[12] C. Guariniello, D. DeLaurentis, Supporting Design via the System Operational Dependency Analysis methodology, Res. in Eng. Des., 28.1 (2017), 53–69.

[13] C. Guariniello, D. Delaurentis, Maintenance and Recycling in Space: Functional Dependency Analysis of On-Orbit Servicing Satellites Team for Modular Spacecraft, AIAA Space Conference and Exposition, 2013.

[14] United States Department of Defense, Systems Engineering Guide for System-of-Systems (2008), http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf (retrieved 10.09.18)

[15] J. Dahmann, G. Rebovich, J. Lane, R. Lowry, K. Baldwin, An Implementers’ View of Systems Engineering for System of Systems, IEEE Systems Conference, 2011.

[16] W. Zhang, Z. Li, W Wang, Q. Li, System of Systems Safety Analysis of GNSS based on Functional Dependency Network Analysis, Appl. Math. Inf. Sci. 10.6 (2016), 2227-2235.

[17] C. Guariniello, D. DeLaurentis, Communications, Information, and Cyber Security in System-of-Systems: Assessing the Impact of Attacks through Interdependency Analysis, Procedia Computer Science, 28 (2014), 720-727.

[18] R. Rolley, R. Potter, S. Zusack, S. Saikia, Life Cycle Cost Estimation of Conceptual Human Spaceflight Architectures, AIAA Space and Astronautics Forum and Exposition, 2017.

[19] P. Garvey, A. Pinto, Advanced Risk Analysis in Engineering Enterprise Systems, CRC Press, 2012.

[20] P. Garvey, A. Pinto, J. Reyes Santos, Modelling and Measuring the Operability of Interdependent Systems and Systems of Systems: Advances in Methods and Applications, Int. J. of System of Systems Eng., 5 (2014), 1–24.

[21] W. O’Neill, D. DeLaurentis, Enhanced Robust Portfolio Optimization for cost, performance risk and schedule analysis of a Lunar mission, 69th International Astronautical Congress, Bremen, Germany, 2018.

[22] W. O’Neill, D. DeLaurentis, Assessing Cost, Performance and Risk of Human Lunar Exploration Missions Using Robust Portfolio Optimization, AIAA Space Forum 2018 Congress, Orlando, USA, 2018.