The following paper will be published and presented at the 2013 IEEE International Systems Conference in Orlando, Florida, 15-18 April, 2013.

SysCon2013 -- IEEE International Systems Conference
Orlando, FL, April 15-18, 2013
Pre-print Version

Considering Alternative Strategies for Value Sustainment in Systems-of-Systems

Nicola Ricci
Systems Engineering Advancement Research Initiative
Massachusetts Institute of Technology
Cambridge, MA 02139
nricci@mit.edu

Adam M. Ross
Systems Engineering Advancement Research Initiative
Massachusetts Institute of Technology
Cambridge, MA 02139
adamross@mit.edu

Donna H. Rhodes
Systems Engineering Advancement Research Initiative
Massachusetts Institute of Technology
Cambridge, MA 02139
rhodes@mit.edu

Matthew E. Fitzgerald
Systems Engineering Advancement Research Initiative
Massachusetts Institute of Technology
Cambridge, MA 02139
mattfitz@mit.edu

Abstract—**Systems of Systems (SoSs) operating in an uncertain world must overcome a variety of challenges in order to sustain value delivery over time. This paper describes strategies for value sustainment, using an application of the "wave model" to represent time-varying SoS Engineering (SoSE) activities and opportunities for SoS-change. A Maritime Security (MarSec) SoS case study is described, and simulation-based Era Analysis is used to evaluate SoS alternatives through different operational environments for an assumed 8-year time frame. Eight SoS designs are evaluated and compared across four strategies in terms of accumulated utility, discounted cost, and total down time. The four value sustainment strategies are: (1) self-recovery, the SoS is not changed (i.e., relating to survivability/robustness); (2) changes in the design of the SoS are allowed (i.e., relating to changeability); (3) changes in the architecture of the SoS are allowed (i.e., relating to evolvability) once, or (4) three times in the eight years. The results provide an example of how quantitative approaches can be used to gain insights into tradeoffs in how SoS architects can create value-sustainable SoSs for the long run.**

Keywords—System of Systems; changeability; evolvability; value sustainment; strategies; architecture;

I. INTRODUCTION

Systems are increasingly more complex, presenting multifaceted challenges for systems architects and decision makers. Over the past twenty years, humankind has experienced changes of great magnitude: the rise of the internet, instantaneous communication and information sharing, disruptive technologies, geo-political shifts, and more. In modern day Systems of Systems, the degree of interconnectedness and interdependence has also increased, leading to increased complexity in making strategic system decisions, as well as making positive changes in one system without introducing the risk of negative impact in other systems. Modern day SoS architects require new strategies for achieving value sustainment.

While a variety of different types of SoSs have been defined in the literature (e.g.: directed, acknowledged, collaborative, virtual) [1], the content of this paper is primarily intended for directed SoS. In directed Systems of Systems, there is a deliberate attempt to create an SoS, and SoS managers exercise control over the constituent systems [1]. The content developed here is also applicable to SoSs that are somewhere between a directed and an acknowledged SoS (one in which the constituent systems retain a high degree of autonomy in their evolution).

II. MOTIVATION

Sustaining stakeholder value delivery in an operational system is a continual and difficult challenge. Unanticipated shifts in stakeholder needs and perturbations to the system can disrupt value delivery [2]. In some cases, the managerial and operational independence [3] of constituents in the SoS creates a situation where there can be contention between the local value desired by constituent system stakeholders and the global value desired for the SoS as a whole. Furthermore, the sheer magnitude and socio-technical complexities of an SoS make changes to an architecture a time- and resource-intensive activity.

Evolving an SoS from a current architecture to a future one (to meet emerging and anticipated needs, often reflecting changes in operational contexts) requires coordination and agreement among the constituents. Yet, at the same time, constituents need to be empowered to make local decisions without constantly consulting all other constituents. Constituent decision makers may have the best intentions of avoiding negative impacts on others, but even so, sometimes, negative impacts can occur.

Most Systems of Systems do undergo periodic re-architecting, but this does not necessarily occur at the "speed of need" [4]. It is realistic to assume that some level of change to the SoS, within constituents and to the SoS as a whole, is continuously ongoing. The question for architects in directed SoSs, who are making system decisions during SoS inception or during a periodic re-architecting activity, is “what value sustainment strategies will be most appropriate?” The answer must include recognizing what changes will prove to be unnecessary, and which necessary changes should occur with greater ease and at less cost.

III. STRATEGIES FOR SOS VALUE SUSTAINMENT

All systems face the uncertainties of operating in dynamic environments with changing stakeholders. For SoSs, there are additional uncertainties, such as uncertainty of constituent system participation [5]. When considering complex systems and SoS, uncertainty can take on different forms and result in a variety of impacts. The system can be affected by exogenous perturbations corresponding to variations in context and expectations [6]. Therefore, it is important that systems architects think of ways to appropriately respond to these perturbations, so that the system can continue to deliver value to stakeholders. For example, increasing UAV (Unmanned Aerial Vehicle) altitude in face of an enemy attack would be a way of changing the operations of the system in order to respond to a disturbance. Likewise, adding new UAV types to an SoS could be one way to leverage emerging technologies, resulting in augmented capabilities. It is important that the possibility of applying such changes is considered during the appropriate stages of design and development of the SoS. Not only should complex systems and SoSs be able to mitigate uncertainty that has a negative impact, but also, when possible, they should be able to intelligently exploit opportunities arising as well.

In recent years, various models and representations have been developed to capture the dynamic nature of SoS Engineering ([7], [8], [9]). The recently developed wave model [9], shown in Figure 1, provides an effective portrayal of the key (ongoing) activities performed by teams of SoS architects and engineers.

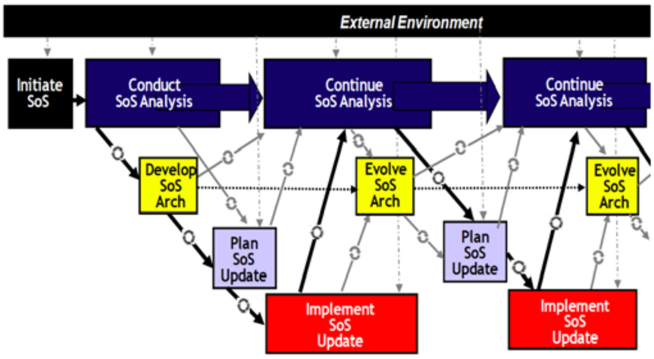


Figure 1. The wave model [9].

In this paper, the wave model representation of SoS Engineering has been used as the basis for illustrating possible responses to perturbations affecting an SoS (see Figure 2). The illustration in Figure 2 emphasizes the perspective – in directed SoSs – for architects to be active agents, continuously controlling (monitoring and acting upon) the state of the SoS. The general structure is analogous to that of a feedback control loop: the SoS can be thought of as a generic system (in operations), and the architects as the sensors (monitoring and analysis) and controllers (imparting changes). Of course, both the “sense” and “control” part of the loop implies consultation and collaboration with engineers and managers of the SoS, as well as the single constituent systems.

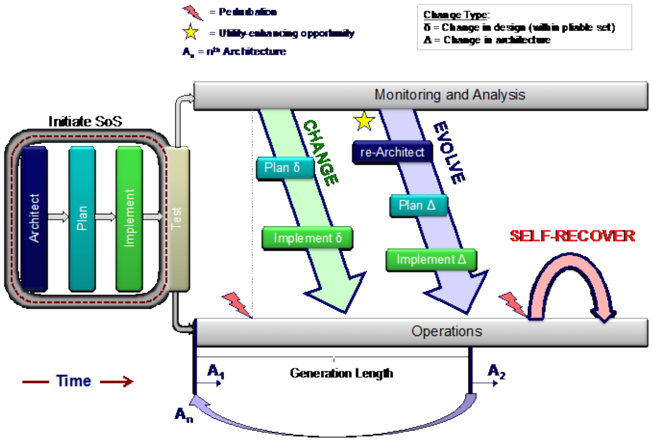


Figure 2. Three different types of responses (change design, evolve architecture, let the system self-recover) illustrated using the wave model.

As systems architects and engineers (among which are designers, operators, maintainers) continuously monitor and analyze the performance of the SoS once it enters the operations phase, they can impart changes in either the architecture or the design of the SoS at any point in time they deem appropriate, subject to their own constraints. In the context of this study, an architecture is defined by a set of characteristics that SoS architects deem *fundamental* to specifying allowed forms, functions and behaviors of the SoS. These characteristics tend to be difficult to change (i.e., high costs and time required, with impact on operations). A design is a specific instance of a given architecture. For example, consider the case of an aircraft carrier: different architectures correspond to the different classes of carriers (Nimitz-class, Ford-class); different designs correspond to different ship designs within a given architecture. USS Nimitz and USS Dwight D. Eisenhower are both Nimitz-class; these ships are instances of the particular aircraft carrier architecture. Changing between design instances within an architecture is usually less challenging and time-consuming than changing from one architecture to another.

In response to perturbations negatively impacting the SoS and opportunities enhancing possible value delivery, architects (working with their team of engineers and SoS managers) can decide to either plan and implement changes in the *design* of the SoS, or to re-architect the SoS (i.e., plan and implement changes in *architecture*). The latter type of changes implies more substantial cost and time, but also has a greater potential impact on value delivery, as illustrated in Figure 3. Alternatively, for the subset of perturbations that negatively affect the SoS, architects can opt for not changing the SoS at all, and letting it recover by itself (if it is able to). Re-architecting activities can also be periodic: architects can decide the time between architecture evolutions (i.e., generation length) of a specific SoS a priori (see Figure 2).

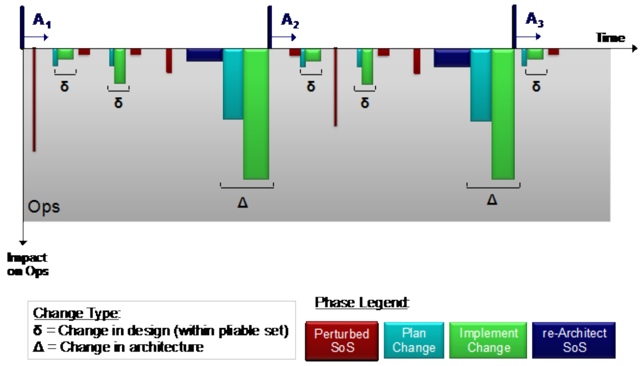


Figure 3. Close up on Operations and the effect of different types of changes in terms of time and impact on SoS operations (value delivery). This figure is purely illustrative.

Based on the foregoing discussion, three possible "change strategies" for value sustainment have been described, corresponding to the three "change" arrows in Figure 2: *change* the design, *evolve* the architecture, and *self-recovery*. These strategies – *change SoS design, change SoS architecture* self-*recovery* – conceptually relate to particular ilities; respectively: *changeability, evolvability, and survivability/robustness.* The choice of which strategy, or mix of strategies, to adopt can have a significant impact on these SoS ilities. Therefore, decisions to be considered for SoS include selecting the initial SoS architecture and design instance, as well as the strategies for altering the SoS over time.

In order to assess and compare the relative value of adopting a given strategy, the performance of the SoS over time should be quantitatively evaluated for the different strategies. The quantitative comparison of alternative strategies for SoS value sustainment over time, i.e. the *type* and *timing* of changes (if any) imparted on the SoS, is performed in this paper using Era Analysis.

IV. ERA ANALYSIS

Era Analysis [10] allows systems architects to generate and investigate system evolution strategies. *Eras* are intended to model potential lifecycles for systems and SoSs by sequencing finite-duration periods of fixed contexts and needs referred to as *epochs*, as pictured in Figure 4. The ability to describe the value of an SoS in a static future context (akin to short run scenario planning) is relatively well understood, so the sequence of epochs provides a functional basis on which to consider lifetime value. As the epochs change over time, the SoS's delivered value changes as well: potentially increasing or decreasing. This concept meshes nicely with the role of the systems architects in Figure 2, as they can observe these changes in context while monitoring the system, and then decide what actions should be taken based on the resulting effect on value delivery. This determination of what response should be made to the outcome of uncertainty is a judgment, and is what is referred to in this paper as a "change *strategy*". As discussed, change strategies can vary from "do nothing", to minor design changes, to architecture-level changes, initiated by different thresholds of acceptable value delivery defined by the architects. It is also worth noting that, when dealing with eras and the progression of the SoS through time, timing is an important variable within the control of the systems architects. In the context of this paper, the era construct is used to assess the SoS performance over time, and to determine the benefit of using a particular value sustainment strategy in the long run.

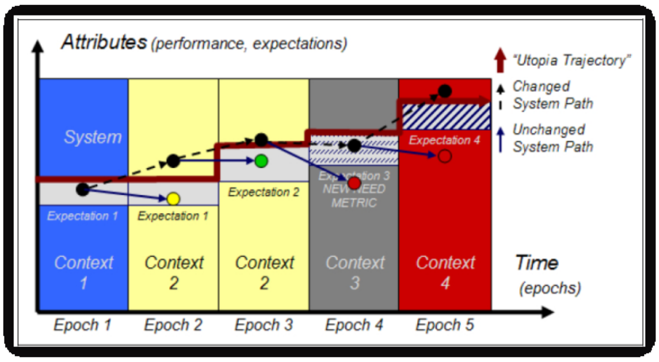


Figure 4. Depiction of a system transversing an Era.

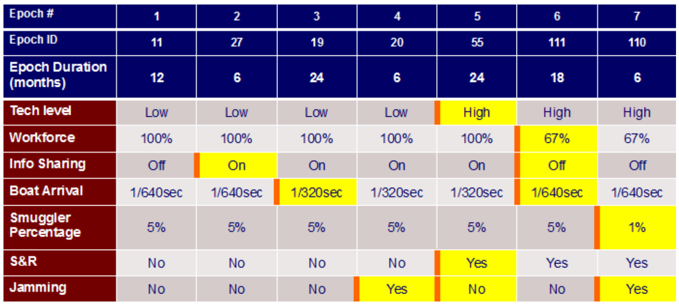
V. MARSEC SOS CASE STUDY

A Maritime Security (MarSec) SoS is considered for exploring the adoption of different "change strategies" over its lifecycle. The main operational goal of the MarSec SoS is to provide maritime security for a particular littoral area of interest. The system is required to detect, identify and board boats that constantly enter and exit the area of interest. Moreover, upon request, it must be capable of providing for search and rescue of sinking boats or entities in danger within the area of interest.

Some of the constituent systems of the SoS are UAVs (two different types), manned patrol aircraft, helicopters, patrol boats, and radar towers. Operational choices include the segmentation of the area (in terms of what is covered by different UAVs), task assignment (what functions are performed by the different constituent systems), and the number of operators per UAV [11]. The design space is defined by the different levels, or states, that the constituent systems and operational choices can have (i.e., 3 vs. 1 Hermes UAVs, or multi-role vs. dedicated task assignment).

This SoS can be considered, to a large extent, a directed SoS, as its managers have full control over the constituent systems (with the exception of, for example, the satellite used for communication relay and Port Authority-managed patrol boats, for which the control is partial or effective under specific circumstances). The performance of the different MarSec SoS design instances was obtained using a discrete-event simulation [12], which allowed for the simulation of various stochastic processes (e.g., boat arrival rate) and relevant emergent behaviors arising from the interaction of constituent systems and operational choices (e.g., how the presence or number of particular UAVs interact with the task management strategy adopted). The simulation evaluated the mission utility and cost of 10,368 alternative design instances across 128 alternative contexts. The simulation model was verified by testing input-output relationships and ensuring that the computer programming and implementation of the conceptual model was correct [13]. Furthermore, in the absence of an experimental or "real-world" example of such an SoS to validate it against, the simulation was subjected to face-validity testing. A visualization tool was developed, and a group of experienced systems architects and analysts (who have been working on a similar Maritime Security SoS for years) used it to confirm that the basic behavior of the SoS, and its response to changes in inputs, were considered to be valid.

The epoch space used for Era Analysis is defined by the various shifts (perturbations that cause shifts in contexts or needs [14]) of relevance for the MarSec SoS case. Some of the shifts considered are: varying the percentage of smugglers in the area of interest, the volume of boats going through the area, and whether or not stakeholders are interested in providing search and rescue capabilities (see Figure 5). For the purpose of performing Era Analysis over the long run for the SoS, a sequence of epochs is defined in order to form an era. An illustrative era, where favorable and unfavorable epoch shifts alternate as the SoS operates through time, has been considered for the analysis of the MarSec SoS, and is encapsulated in the table in Figure 5. In this table, the change in the level of a given epoch variable (row labels, in red) causes a change in epoch (a shift), and is highlighted in yellow. The epoch ID and duration are also included.



Epoch #	1	2	3	4	5	6	7
Epoch ID	11	27	19	20	55	111	110
Epoch Duration (months)	12	6	24	6	24	18	6
Tech level	Low	Low	Low	Low	High	High	High
Workforce	100%	100%	100%	100%	100%	67%	67%
Info Sharing	Off	On	On	On	On	Off	Off
Boat Arrival	1/640sec	1/640sec	1/320sec	1/320sec	1/320sec	1/640sec	1/640sec
Smuggler Percentage	5%	5%	5%	5%	5%	5%	1%
S&R	No	No	No	No	Yes	Yes	Yes
Jamming	No	No	No	Yes	No	No	Yes

Figure 5. Table describing the sequence of epochs that form an era of average difficulty, where contexts fluctuate between good and bad.

In the context of the MarSec SoS, four "change strategies" are considered: (1) *no changes allowed*; (2) *only changes in the design of the SoS allowed*; (3) *a single change in the architecture is allowed*; and (4) *three changes in architecture are allowed*. For strategies (3) and (4), changes in the design are allowed as well.

Architecture-related changes are assumed to be those involving changes in the number of zones considered for UAV coverage of the area of interest, the authority type (central vs. distributed), whether or not to include a workforce buffer, and the addition of vehicles to the SoS. Such changes are permissible only at the initial or re-architecting points in the lifespan of the SoS. All other changes in the number of constituent systems and operational choices (task assignment and number of operators per UAV) are considered changes in the design of the SoS. Such changes are permissible at any time, as they do not impact the architecture.

The possible types of changes considered for the MarSec SoS case can be summarized as follows:

1. Reduce to pre-validated vehicle set (design-level change), i.e. expert-picked stable designs, which can be implemented with current operating SoS constituent systems.
2. Changing short-term (design-level change) ConOps (i.e., Task Assignment, Operators per UAV).
3. Changing long-term (architecture-level change) ConOps (i.e., number of Zones, Authority).
4. Adding Constituent Systems (architecture-level change). The cost of adding is the cost of the new vehicles, and the delay of adding is vehicle-dependent.

Changes in design follow a "maximize efficiency" change-execution rule, which is: when a perturbation hits the SoS (causing an epoch shift), systems architects change the design of the SoS so that it moves as close to the Pareto frontier of the utility vs. cost tradespace as possible [15]. It is important to note here that it is possible to choose any other preferred change-execution rules (e.g., maximize utility, minimize cost) [15]. The Pareto frontier is the locus of designs with maximum efficiency, where improving on a given objective (e.g., utility) necessarily induces a worse score in the conflicting objective (e.g., cost). The re-architecting schedules for strategies (3) and (4) are assumed to be fixed: after four years in the case of one re-architecting, and after two, four and six years in the case of three re-architecting activities. Figure 6 illustrates the four strategies using the basis of the wave model representation shown in Figure 2.

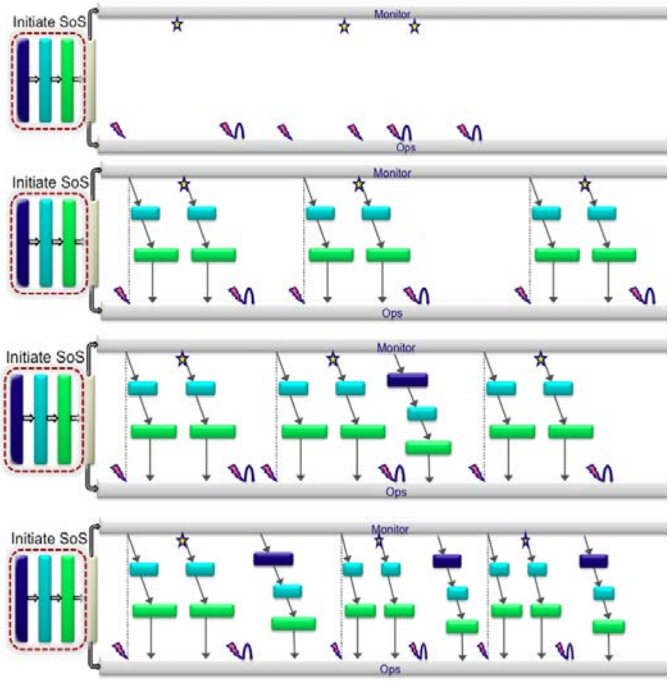
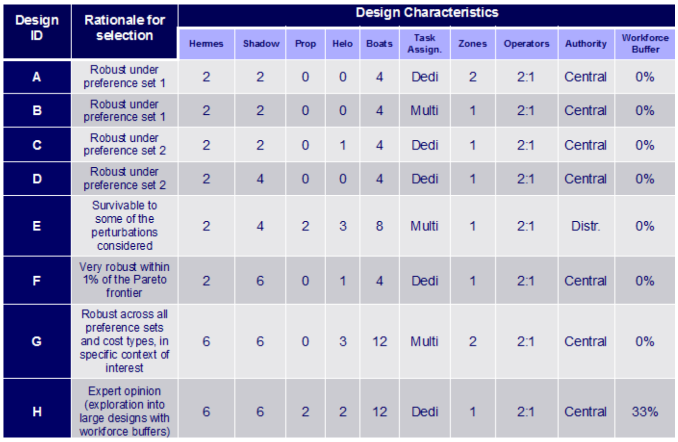


Figure 6. Four "change strategies": no changes (top), changes in design only (second from top), one re-architecting (third from top), and three re-architecting (bottom). The "lighting bolts" illustrated in this figure correspond to epoch shifts (see Figure 5).

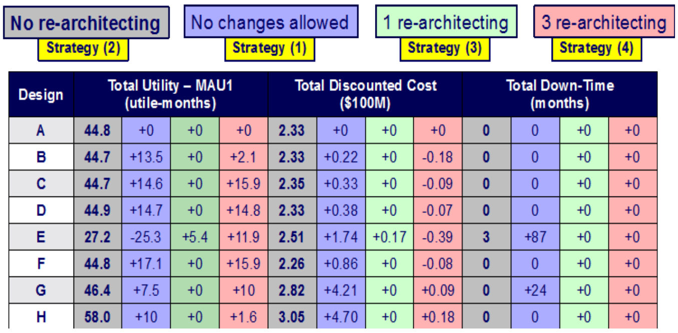
Era analysis is used for the evaluation of the lifecycle performance of eight SoS designs of interest, each of which is associated with a specific initial architecture. These designs have been previously selected using robustness and survivability screening metrics [15], and are presented in the table in Figure 7.



Design ID	Rationale for selection	Design Characteristics									
		Hermes	Shadow	Prop	Helo	Boats	Task Assign.	Zones	Operators	Authority	Workforce Buffer
A	Robust under preference set 1	2	2	0	0	4	Dedi	2	2:1	Central	0%
B	Robust under preference set 1	2	2	0	0	4	Multi	1	2:1	Central	0%
C	Robust under preference set 2	2	2	0	1	4	Dedi	1	2:1	Central	0%
D	Robust under preference set 2	2	4	0	0	4	Dedi	1	2:1	Central	0%
E	Survivable to some of the perturbations considered	2	4	2	3	8	Multi	1	2:1	Distr.	0%
F	Very robust within 1% of the Pareto frontier	2	6	0	1	4	Dedi	1	2:1	Central	0%
G	Robust across all preference sets and cost types, in specific context of interest	6	6	0	3	12	Multi	2	2:1	Central	0%
H	Expert opinion (exploration into large designs with workforce buffers)	6	6	2	2	12	Dedi	1	2:1	Central	33%

Figure 7. List of the eight SoS designs used for the analysis. The rationale behind the choice and the design characteristics are presented in the table.

The previously described simulation – enabling the evaluation of the performance of different MarSec SoS designs against dynamic contexts and varying stakeholders needs (i.e., epochs) – is used for the application of Era Analysis [10]. Depending on the strategy adopted, systems architects can choose to transition to a new design (and possibly architecture) at any desired point in time (see strategies (3) and (4)), or, alternatively, every time there is an epoch shift (see strategy (2)). The Era Analysis results for the four value sustainment strategies employed are compared in terms of three different metrics: total utility, i.e. the total utility accumulated by the SoS throughout its lifecycle, measured in utile-months; total discounted cost, i.e. the total cost of designing and operating the SoS over its lifecycle (discounted); and the total down-time, i.e. the time (in months) in which the SoS is down and not delivering value. The first two of these metrics can be used to derive affordability-related considerations. Figure 8 shows the results obtained for the era described in Figure 5. In this analysis, strategy (2) (no re-architecting – only changes in design allowed) is taken to be the baseline, and the results of all other strategies are compared to it.



No re-architecting — Strategy (2)
No changes allowed — Strategy (1)
1 re-architecting — Strategy (3)
3 re-architecting — Strategy (4)

Design	Total Utility – MAU1 (utile-months)				Total Discounted Cost ($100M)				Total Down-Time (months)			
A	**44.8**	+0	+0	+0	**2.33**	+0	+0	+0	**0**	0	+0	+0
B	**44.7**	+13.5	+0	+2.1	**2.33**	+0.22	+0	-0.18	**0**	0	+0	+0
C	**44.7**	+14.6	+0	+15.9	**2.35**	+0.33	+0	-0.09	**0**	0	+0	+0
D	**44.9**	+14.7	+0	+14.8	**2.33**	+0.38	+0	-0.07	**0**	0	+0	+0
E	**27.2**	-25.3	+5.4	+11.9	**2.51**	+1.74	+0.17	-0.39	**3**	+87	+0	+0
F	**44.8**	+17.1	+0	+15.9	**2.26**	+0.86	+0	-0.08	**0**	0	+0	+0
G	**46.4**	+7.5	+0	+10	**2.82**	+4.21	+0	+0.09	**0**	+24	+0	+0
H	**58.0**	+10	+0	+1.6	**3.05**	+4.70	+0	+0.18	**0**	0	+0	+0

Figure 8. Results from application of era analysis for 8 different SoSs and 4 different "change strategies".

Both general and specific insights can be derived from these results:

- For strategy (2), there is similar utility performance amongst designs of interest, except for design E (poor) and H, which is excellent, but at the expense of a 25% increase in cost.
- Strategy (1) focusing on *survivability* and *robustness* puts some SoS (E and G) at a higher risk of downtime; this is because the inability to change does not let the SoS bounce back after a perturbation, and therefore it no longer delivers sufficient value to stakeholders.
- Performing design-level changes only (i.e., strategy (2) – *changeability*) yields better results for relatively small SoS; a larger initial SoS has less "room" for change and higher operational costs. This is apparent when going from strategy (1) to strategy (2), as the tradeoff appears likely to be most beneficial for small designs (i.e. designs A, B, C, D) with approximately +25% utility, and +10% cost, but less so for larger designs (i.e. designs E, G, H), with approximately +15% utility, and +100% cost.
- In going from strategy (2) to (3), only one design draws benefit, but in going from strategy (2) to (4), nearly all designs are benefitted. This demonstrates the significant effect that timing can have on the value of a re-architecting effort. For this era, the fixed re-architecting after 4 years has little effect, while those at 2 and 6 years are more valuable, but there is no way to know this in advance until the uncertainty resolves. This inspires the question of how systems architects could plan for flexible re-architecting schedules in order to exploit this behavior, and is a promising topic for future research.
- In going from strategy (2) to (1), all designs but E are benefitted in terms of utility, but have a higher cost. This is due to the nature of the "maximize efficiency" changing strategy applied for strategy (2), which for the MarSec SoS design space tends to favor small, low-cost designs.
- All designs (except for A) benefit from the triple re-architecting, even to the point of gaining utility and decreasing cost simultaneously.

Finally, it is also important to realize that different strategies can be considered, as well as alternative eras. The results obtained in this analysis are relative to the assumptions the analyst makes in terms of what could possibly happen over time (i.e., nature and probability of occurrence of epochs) and what changes he or she can make to the SoS. In other words, a general understanding of the contextual and needs-related uncertainties affecting the SoS, as well as the possible strategies one can employ to respond to how uncertainty unfolds, are prerequisites for the application of the analysis presented. Through the analysis, important decisions in terms of what value sustainment strategies are most appropriate under what conditions can be deduced.

VI. DISCUSSION AND CONCLUSION

The analysis performed, although demonstrative in nature, addresses some of the important issues that characterize modern-day SoS Engineering. To begin with, it stressed the importance of considering (and possibly highlighting options to facilitate the implementation of) possible timed and/or contingent SoS changes early in the design phase. Often times, in fact, it is difficult to design and implement a change to the SoS while it is in its operations phase, and having previously identified options [16] that can facilitate the implementation of such changes can turn out to be a key decision to sustain the performance of the SoS. For example, in the case of MarSec, the loss of a detection UAV can result into significantly lower performance (especially for SoSs with small initial UAV fleets). A possible response to such a loss can be a reorganization of the roles of the remaining UAVs and other assets (i.e., change in task assignments), in order to resume acceptable performance. Such a change may be difficult to implement while the system is operating, as it implies the addition of extra capabilities (for detection, in this case) into operating assets, cooperation among the workforce to coordinate and perform contingent tasks, as well as potential loss in performance if the system requires down time (i.e., the time it takes to design and implement the change). However, if "multi-role" assets had been already included in the SoS and the workforce was initially trained to perform multiple tasks, the implementation of this change to the SoS could become less resource-intensive. The importance of the existence of pre-conceived (and pre-enabled) change strategies is demonstrated in the analysis by design E, which, for example, performs very poorly in the absence of the possibility to rapidly and effectively change.

Furthermore, the analytical approach highlights how the value of a given strategy (self-recover, change or re-architect) is dependent on the initial design selected. For example, the analysis showed that the "change" strategy involving the prompt addition of back-up assets to the operating SoS is more effective for smaller (in number of assets) initial SoS designs. Lastly, the analysis allowed for consideration of the timing of execution of architectural changes. While re-architecting one time was hardly beneficial for any design, when re-architecting three times, nearly all designs experienced improved performance. This kind of analysis allows systems architects to investigate what the most efficacious re-architecting time-scales are, as well as inspires the question of how they could plan for flexible re-architecting schedules.

Even though this paper only sets up a demonstration analysis – and future work will concern the validation of the approach – the application of Era Analysis can enable the exploration and comparison of results associated with the adoption of alternative "change strategies" for different SoS designs initially implemented. In this regard, it can help systems architects choose an appropriate value sustainment strategy for a given envisioned era, as well as highlight salient time-scales for SoS intervention (e.g. when to go through the effort of "re-architecting" and when to build in "options" for shorter time-scale changes to the SoS). This type of analysis can also identify changes that will turn out to be unnecessary, or others that will prove to be very important and need to occur with greater ease. Moreover, this approach invites systems architects to think about the adoption of different strategies before the inception of the operational SoS. The foreknowledge of possible value sustainment strategies can facilitate coordination and agreement among the constituent systems for the implementation of potential future changes in the SoS. This way, imparting changes to the SoS can become a less time and resource intensive procedure. It is important to note that the strategy selection for a particular design is dependent on the era considered, and that a "multi-era" analysis could be the basis for future research in terms of helping architects to select design-strategy pairs that are most robust across different likely era progressions.

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding for this research provided through MIT Systems Engineering Advancement Research Initiative (SEAri, http://seari.mit.edu) and its sponsors.

REFERENCES

[1] J. K. Bergey, S. Blanchette Jr., P. C. Clements, M. J. Gagliardi, J. Klein, R. Wojcik, and B. Wood, "U.S. Army Workshop on Exploring Enterprise, System of Systems, System, and Software Architectures", Technical Report, CMU/SEI-2009-TR-008, ESC-TR-2009-008, SEI Administrative Agent. Copyright 2009 Carnegie Mellon University.

[2] B. Mekdeci, A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Examining Survivability of Systems of Systems," INCOSE International Symposium 2011, Denver, CO, June 2011.

[3] M. W. Maier, Architecting Principles for Systems of Systems. In Systems Engineering 1 (4): pp. 267-284.

[4] S. J. Hutchison, "Test and Evaluation at the Speed of Need", Document of Defence Technical Information Center (DTIC), May 2010.

[5] D. Chattopadhyay, A. M. Ross, and D. H. Rhodes, "Combining Attributes for Systems of Systems in Multi-Attribute Tradespace Exploration", 7th Annual Conference on Systems Engineering Research 2009 (CSER 2099).

[6] A. M. Ross, H. L. McManus, A. Long, M. G. Richards, D. H. Rhodes, and D. E. Hastings, "Responsive Systems Comparison Method: Case Study in Assessing Future Designs in the Presence of Change", AIAA Space 2008, San Diego, CA, September 2008.

[7] DoD, "Systems Engineering Guide for System of Systems", Version 1.0. Office of the under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), 2008.

[8] B. Boehm, and J. A. Lane, "Using the Incremental Commitment Model to Integrate System Acquisition, System Engineering, and Software Engineering", Journal of Defense Software Engineering, October 2007.

[9] J. Dahmann, G. Rebovich, R. Lowry, J. Lane, and K. Baldwin, "An Implementers' View of Systems Engineering for Systems of Systems", Systems Conference (SysCon), 2011 IEEE International, pages 212-217.

[10] M. E. Fitzgerald, and A. M. Ross, "Sustaining Lifecycle Value: Valuable Changeability Analysis with Era Simulation," 6th Annual IEEE Systems Conference, Vancouver, Canada, March 2012.

[11] B. Mekdeci, A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Investigating Alternative Concepts of Operations for a Maritime Security System of Systems," INCOSE International Symposium 2012, Rome, Italy, July 2012.

[12] B. Mekdeci, Managing the Impact of Change through Survivability and Pliability to Achieve Viable Systems of Systems, *Doctor of Philosophy Dissertation*, Engineering Systems Division, MIT, February 2013.

[13] R. Sargent, "Verification and Validation of Simulation Models", Proceedings of the 1998 Winter Simulation Conference.

[14] J. C. Beesemyer, Empirically characterizing evolvability and changeability in engineering systems, *Master of Science thesis*, Aeronatics and Astronautics, Massachusetts Institute of Technology, June 2012, pp. 51-56.

[15] M. E. Fitzgerald, A. M. Ross, and D. H. Rhodes, "Assessing Uncertain Benefits: a Valuation Approach for Strategic Changeability (VASC)," INCOSE International Symposium 2012, Rome, Italy, July 2012.

[16] A. M. Ross, and D. H. Rhodes, "Anatomy of a Change Option: Mechanisms and Enablers", SEAri Working Paper WP-2011-1-2, http://seari.mit.edu/papers.php [cited 03-01-2013].