Combining Attributes for Systems of Systems in Multi-Attribute Tradespace Exploration

Debarati Chattopadhyay[1] and Adam M. Ross [2] and Donna H. Rhodes [3]

[1]Massachusetts Institute of Technology, USA, debarati@mit.edu
[2]Massachusetts Institute of Technology, USA, adamross@mit.edu
[3]Massachusetts Institute of Technology, USA, rhodes@mit.edu

Abstract

One of the principal value propositions for the creation of Systems of Systems (SoS) is the ability to generate stakeholder value beyond that which can be delivered by a single system or even a collection of systems. The dynamic interactions among the component systems in a SoS make conceptual design decisions for SoS more complex compared to traditional system design. There are several heuristics and qualitative guidelines for designing SoS in the literature, but there is a lack of practical quantitative methods for SoS concept exploration. Development of quantitative methods for SoS conceptual design will greatly improve the ability of decision makers to select SoS designs in the concept design phase that will be value robust over time, by allowing them to consider a larger and more complete set of possible alternative SoS designs than is possible with qualitative methods alone. Multi-Attribute Tradespace Exploration has been used in the past to compare large numbers of system alternatives on a common cost-utility basis. In this method, the designer elicits the decision maker's needs and formulates these as quantified attributes. The systems are then analyzed in terms of their ability to achieve the desired levels of attribute metrics. SoS-level attribute calculations must reflect component system interactions and emergent SoS-level value, as well as the added costs and benefits in the SoS as compared to that of the component systems operating alone. This paper introduces techniques for the SoS attribute combination modeling within the Multi-Attribute Tradespace Exploration method when considering SoS with heterogeneous component systems. Combining the attributes for the SoS-level must take into account the nature of the attributes provided by the component systems as well as the concept of operations for SoS. The techniques used for SoS attribute combination are classified according to the level of coordination between the component systems in the SoS. Using the information about the complexity of the method required along with knowledge about the control structure in the SoS, the SoS architect can estimate the added SoS integration costs. The ability to combine attributes contributes an essential constituent for the extension of the tradespace exploration methodology from the single system to the SoS level, allowing for the comparison of many SoS alternatives on a common cost-utility basis.

Keywords: System of Systems, tradespace exploration, conceptual design

1 Introduction

Systems of Systems (SoS) are higher order complex systems that are composed of independent component systems. They are of increasing importance as today's organizations use highly networked but independently managed systems to generate unique capabilities beyond that available by independent operation of the component systems. Examples are multi-modal transportation networks[1], the Army Future Combat System [2], networks of computers, or a surveillance network consisting of multiple diverse assets. Designing such complex systems with multiple independent components and the interfaces between them requires significant system engineering activities on the part of the SoS designer. The needs of multiple SoS stakeholders as well as component system stakeholders must be considered when designing an SoS with independently managed components. This leads to a complex stakeholder value proposition for the SoS and the need to explicitly consider the relationship between the SoS designer and the component system management, in order to determine the feasibility of an SoS design given uncertain component system participation. Based on the degree of control a SoS designer has over the component systems, SoS can be classified into directed, acknowledged, collaborative and virtual SoS [2], with each category providing limits on the design control available to the designer. Additionally, the use of both legacy and clean sheet component systems in the SoS design results in varying degrees of design control for the SoS designer over the component systems. Interactions between the component systems as well as the time-varying composition of the SoS are also factors in creating a complex design problem requiring advanced system engineering capabilities.

2 Motivation

SoS have added complexity compared to traditional systems both in the concept exploration phase as well as throughout the system lifetime, due to the managerial independence of component systems as well as the dynamic, time-dependent composition of the SoS. To manage this complexity, SoS require continual intensive decision making during all phases of the system engineering lifecycle. Typically, a large proportion of a system design's resources are allocated during the conceptual design phase, and thus it is especially important to make decisions in this phase which result in SoS designs that maintain value over the entire system lifetime. There are several heuristics and guidelines that have been proposed in the literature to help designers during the conceptual design phase of SoS design [3, 4]. However, there are few methods that allow the quantitative comparison of SoS architecture alternatives. A quantitative method of SoS concept design will enable the comparison of a much larger number of SoS designs than is possible with heuristics, leading to a clearer understanding of the key tradeoffs available. There is a need for a quantitative SoS conceptual design framework that will provide decision makers with the ability to compare diverse, multi-concept SoS on the same basis as well as the means to select SoS designs that are likely to be value robust over a number of changing context scenarios. In this paper, a quantitative method for SoS tradespace exploration is introduced to meet the above needs and the key aspect of SoS attribute modeling is described in detail.

3 The SoS Tradespace Exploration Method

Tradespace exploration can be used in system conceptual design in order to compare many different system architectures on the same performance and cost basis. Dynamic Multi-Attribute Tradespace Exploration (Dynamic MATE) is a method for system concept exploration that incorporates tradespace exploration as well as enables the study of effects of dynamically changing contextual variables such as stakeholder preferences, changes in requirements, availability of assets, etc. [5]. Using an approach from Dynamic MATE called Epoch-Era Analysis, many different potential contexts can be simulated, and designs that are value robust over a variety of future contexts can be identified. Dynamic MATE thus enables value robustness considerations early in the conceptual design phase, and through the quantitative comparison of many different system designs, provides the system designer with a large amount of information on the basis of which the critical resource-allocating decisions can be made. As an SoS conceptual design method must necessarily include the means to compare diverse multi-concept SoS architectures on the same performance and cost basis, as well as the ability to consider a large number of future scenarios representative of the dynamic SoS lifetime, Dynamic MATE, which already demonstrates some of these features, is a suitable conceptual design method that can be extended to the SoS domain. However, several enhancements to the Dynamic MATE method are required in order to create a conceptual design method that can deal with SoS-specific design issues, such as the inclusion of both legacy and new components in SoS designs, the multi-level stakeholder value proposition that requires the consideration of SoS-level and component-level stakeholders, and the dynamic composition of the SoS due to entry and exit of SoS component systems, as described in [6]. The steps in the new SoS tradespace exploration method are shown in Figure 1.

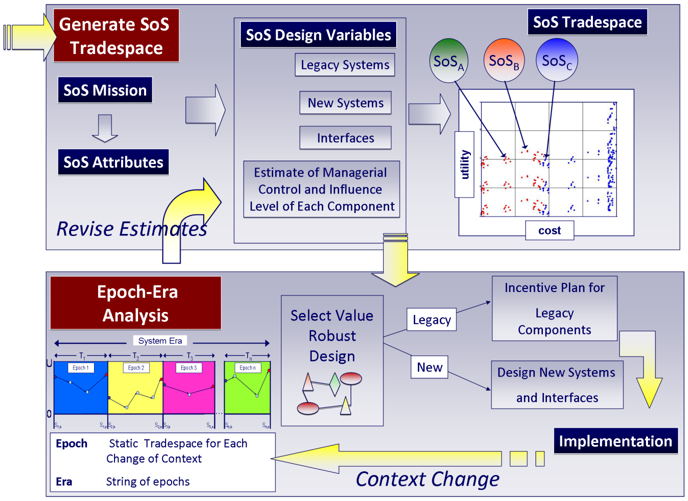

Figure 1 – *SoS Tradespace Exploration Methodology*

The SoS tradespace exploration methodology is intended for use in SoS design during the concept exploration phase. In this methodology, the starting point is the definition of the SoS mission, as provided by the SoS stakeholders. The user-defined SoS performance attributes are obtained through interviews of the SoS stakeholders, and are used to measure the value delivery of the SoS designs. At this point in a traditional conceptual design method, potential concepts would be generated and decomposed into their design variables. However, for the SoS, the generation of concepts corresponds to creating a list of potential legacy and new component systems that may be used to accomplish the SoS mission. The SoS 'design variable' set includes these component systems, their method of interaction and interfaces, and an estimate of the 'participation risk' of each component system in the SoS. The participation risk represents the risk that a component system that is required for the SoS operation will not be available to the SoS, and is related to the relationship between the SoS designer and the component system management. SoS designs can be generated by combining component systems from the list of possible systems, and the performance and cost for each design can be modeled using the corresponding SoS design variables. Utility can be calculated using a multi-attribute utility function, as has been used in previous MATE studies[7]. Using this utility function and the model-derived cost, many diverse SoS designs can be represented on a tradespace. This step of the method can be repeated for a variety of future contexts using an analysis method called Epoch-Era Analysis [8], resulting in a collection of tradespaces representing the SoS performance in different scenarios. Tradespace statistics can be generated over a large number of tradespaces thus obtained, and value robust designs can be identified as the designs that are provide high benefit at low cost for many future contexts[9]. Identified value robust designs can be carried on to the next stage of conceptual design, where more detailed analysis and design can be conducted for new component systems, or interfaces between legacy systems. As this methodology can be quickly repeated, iterations of the method can be done for new changes in context, or when additional detailed information becomes available.

4 Combining Attributes to Model SoS Value

An integral part of the development of this method is the ability to estimate, in the concept exploration phase, the value generated by potential SoS designs. The SoS system value delivery is a function of the component systems' value delivery, but is also greater than just the aggregation of the component system performance. Within the MATE framework, system attributes are decision maker perceived metrics used to measure the system value delivery to stakeholders.

This paper proposes the use of component system attributes, as well as information about the concept of operations of components within the SoS, to model the SoS performance attributes. The classification of the component system attributes and the level of complexity of combination of the attributes is used to estimate the SoS integration cost.

In this section, descriptions are given of the different classes of attributes and the levels of combination complexity of attributes, and then their incorporation into the quantitative SoS tradespace exploration method is explained.

4.1 Attribute Classes

According to [10], system attributes can be classified on the basis of whether they are articulated by the decision maker, as well as the cost to 'display' the attribute in the system. The attribute classifications are shown in Table 1 below.

Class	Name	Property of Class	Cost to Display
0	Articulated Value	Exist and assessed	0
1	Free Latent Value	Exist, but not assessed	0
2	Combinatorial Latent Value	Can exist by recombining class 0 and 1 attributes	Small
3	Accessible Value	Can be added through changing the design variable set (scale or modify system)	Small → large
4	Inaccessible Value	Cannot be added through changing design variable set (system too rigid)	Large → infinite

Table 1 – *Attribute Classification from [10]*

Articulated attributes are attributes that are communicated by a system decision maker to the system designer as part of the system needs. The system performance is assessed on the basis of these articulated attributes. 'Existing' attributes are those that the system exhibits by design.

Class 0 attributes are those that are traditionally considered in system design. The system value delivery to local system stakeholders is measured using the Class 0 attributes, but this is usually only a subset of the possible attributes displayed by the system. The system may also display attributes that are not articulated by any system stakeholder, and represent potential free latent value in that system. This additional latent value comprise Class 1 attributes that can be utilized at no cost by the decision maker, as they are already displayed by the system. Class 0 and Class 1 attributes (together, the total existing value delivered by the system) may often be further enhanced at some small cost to obtain combinatorial latent value, or Class 2 attributes. A re-specification of displayed attributes is often all that is required to access information that is already available from the system, and the system may or may not need to be modified to obtain Class 2 attributes. Beyond these three classes of attributes, Class 3 attributes are those achievable by modification of the original system at some cost to obtain new performance characteristics. Class 4 attributes are only available through drastic changes of the original system at prohibitive cost, or not achievable at all given the physical constraints of the original design.

SoS system components have some level of independent management, and thus each have a local set of stakeholders. The Class 0 attributes of each component system derives from this set of stakeholders, and thus may not (more likely, will not) coincide exactly with the required SoS attributes, obtained from the SoS stakeholder set. However, from the above discussion, it is evident that for a particular SoS to satisfy its system goals, the SoS attributes must be a subset of the combined set of Class 0, 1, 2 and 3 attributes of the component sets. As Class 0 and 1 attributes are more readily available to the SoS designer than Class 3 attributes, which require modification of or addition to the systems at some cost, SoS attributes that utilize Class 0 and Class 1 attributes of the component systems are more easily achievable compared to those that require Class 3 component system attributes.

Given that the SoS attributes are a subset of the Class 0, 1, 2, and 3 attributes of the SoS components, generating the SoS attribute values requires some combination of the component attributes. The method by which the component attributes are aggregated is determined by a classifying the 'level of combination complexity' selected for the particular attribute, described in the following subsection.

4.2 Level of Attribute Combination Complexity

The way in which the SoS components interact during SoS operation is an integral consideration in determining the interfaces between the components and also in determining how component attributes can be combined to achieve the SoS attributes. In early concept exploration, there are few constraints on the design space, so the SoS designer needs to explore the possible methods of attribute combination to determine which type is suitable for the particular SoS design. As this space of possibilities is large, as a first-order estimate, three types of attribute combination methods can be defined: low level, medium level and high level complexity. It is assumed that the higher the complexity of attribute combination, the higher the cost for creating an interface capable of this kind of combination. Low level methods involve taking the best performance in a particular SoS attribute from the set of components in the SoS. If the SoS attributes are such that the mission is differentiated between the components (i.e. each component system provides a unique subset of attributes), then this is the level of attribute combination that is required . There may be cases where component systems operate in parallel, but only the best performance is chosen - such as when a SoS component with high performance is only available to deliver that attribute for part of the operating time. An interface between the components that generates an attribute through this level of combination will be relatively low complexity and therefore lower cost.

Medium level attribute combination is required when there are more complex SoS concepts of operation. For example, when there is a handoff between different assets in the SoS, such that multiple components are involved in delivering a single attribute performance, the resulting SoS attribute performance is a combination of the two component attribute performances. Methods used for SoS attribute combination at the medium level may involve time-weighted averaging, such as the concept of time-weighted average utility [11]. Due to the higher complexity of operations in this case, the additional costs required to create interfaces enabling operations must be carefully considered.

The highest level of attribute combination is required when multiple SoS components deliver performance relating to the same SoS attribute simultaneously. In this case, fusion of the attributes at a more detailed level than just averaging is required. A possible set of methods for combination of attributes at this level is data fusion. Data fusion is a well-developed field with methods available for combining data-related attributes, such as image resolution. Data fusion is defined as the combining of data from multiple sources along with database information to draw inferences beyond that obtainable through a single data source alone [12]. Within data fusion, the DoD Joint Directors of Laboratories (JDL) data fusion levels represent multiple levels of complexity that result in different amounts of information obtained from a set of data [13]. The SoS designer can select a particular data fusion method for attribute combination in order to obtain the SoS attribute from multiple component sensors. Within this high level of attribute combination, the JDL levels of data fusion may provide additional information about the complexity of attribute combination, and thus help refine the additional cost of combining the component attributes for the SoS.

As an example of the levels of attribute combination required in an SoS, consider a multi-modal surveillance SoS consisting of a satellite radar asset and an aircraft radar asset, illustrated in Fig 2. In the case where there is no overlap between the SoS relevant attribute sets required from each asset - the satellite performs imaging and the aircraft performs target tracking activities - the attribute combination method simply consists of taking the attribute performance of each component and representing it as the SoS attribute. There may be additional SoS-level attributes - such as responsiveness of the SoS (i.e. how quickly after a request coverage can be obtained of a particular area of interest). In the case of the SoS responsiveness attribute, the best performance betwen the satellite and the aircraft is an appropriate SoS performance measure. This would be a scenario in which low level attribute combination is sufficient to model the SoS value delivery.

Continuing with the same surveillance example, suppose the concept of operations was changed such that the satellite would identify a target in its field of view, notify the aircraft, and then track the target until the air asset arrived and the track could be handed off. In this case, an SoS attribute such as track life - the length of time for which the SoS can track an object of interest after successful identification - would be a combination of the two track life attribute performances of the component systems. Similarly, an attribute like image resolution could also be represented as a time-weighted average of the component system resolutions during the above operation. This SoS operating scenario would be a candidate for medium level attribute combination, with some associated cost for the implementation of the interface.

If the satellite and aircraft operate simultaneously, imaging overlapping fields in different wavelengths for example, detailed data fusion may be required. JDL Level 0 or Level 1 processing of data [14] may be suitable to help identify targets in the multi-wavelength images, providing faster target identification than with a single asset. However, complex interfaces and additional component systems may be required in order to achieve the data fusion.

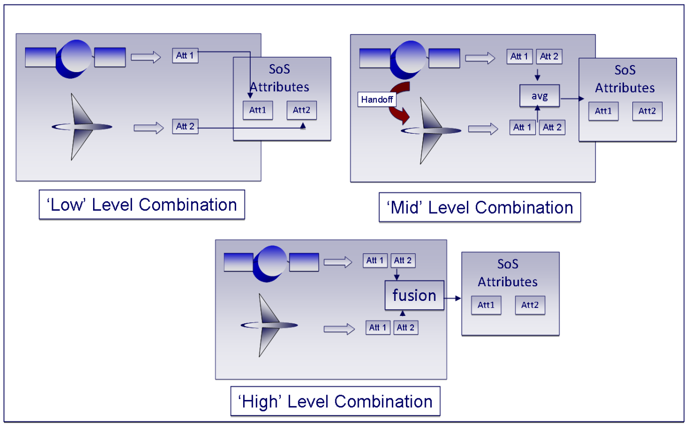

Figure 2 – *Levels of Attribute Combination Complexity Example*

4.3 Effects of Level of Attribute Combination and Attribute Classes on the Tradespace

From Sections 4.1 and 4.2, it is evident that the SoS-level attributes and costs are influenced by both the attribute class of the attributes combined, and the level of combination complexity used to merge those attributes.

The level of attribute combination required to attain a particular SoS attribute determines the method used to combine the component attributes. Once this method is selected, the SoS attribute value is obtained as a function of the component system attributes.

$$\textit{SoS attribute value} = \textit{fn (componentA attribute, componentB attribute, componentC attribute,..)} \quad (1)$$

On the basis of the SoS attribute value thus calculated, the SoS single attribute utility is calculated within the MATE framework, using the defined stakeholder utility preferences. The single attribute utilities calculated for each attribute are then combined using Multi-Attribute Utility Theory to obtain a SoS multi-attribute utility which is used to compare various designs on a tradespace.

The attribute class information for each component attribute involved in the SoS attribute calculation is used to estimate the additional cost required to create the SoS. At the early stages of concept exploration, the SoS designer is primarily concerned with determining a set of component systems that are suitable to integrate into the SoS. The interfaces are not yet defined as the composition of the SoS is undetermined. Thus an estimate is needed for the additional cost required to unite the independent component systems into a SoS. The class of a component attribute and the level of complexity selected are an indication of the 'integration cost' required to combine that attribute. Combining two component attributes that are Class 0 will require less effort and cost than combining two attributes that are Class 3. The additional cost for a Class 3 attribute versus a Class 0 or Class 1 attribute is determined by the SoS designer, perhaps in consultation with the component system management. For the purposes of the example in this paper, an additional cost of 10 percent of the component cost for Class 2 attributes and 30 percent of the component cost for Class 3 attributes is used. These are estimates that will need to be refined on a case by case basis for particular SoS as well as for particular components within the SoS.

Considering each attribute in the SoS attribute list, if higher class component attributes are required to attain the SoS attribute, the resulting additional cost is added to the total SoS cost estimated though other means (such as from the component system costs). This total estimate for cost thus encompasses the difficulty and expense required to integrate the SoS components. This method for estimating SoS cost enables the comparison on the same basis of different SoS designs with different component compositions and includes the consideration of difficulty of achieving the final configuration.

4.4 Participation Risk

Chattopadhyay [6] discusses the idea of 'effective managerial control' - the combination of managerial control [3] and influence that results in a metric representing the ability of the SoS designer to affect the behavior of a component system. Changes in component system behavior may include the components' joining or leaving the SoS, or changes to the design of the component leading to altered component performance characteristics. The idea of control is clearly important in all SoS considerations as it affects the ability of the SoS designer to obtain expected performance from the component systems. In the specific case of attribute combination, the availability of Class 3 attributes and different levels of attribute combination is dependent on the level of cooperation between component systems in the SoS, as well as the distribution of costs - both related to effective managerial control.

Participation Risk (PR) of a component system is defined in [6] as the gap between certain participation of a required SoS component system and the effective managerial control of the SoS designer over that component. Thus PR can be considered a measure of the uncertainty associated with attaining a particular desired SoS design configuration. In the SoS tradespace, this can be represented as a third dimension, along with utility and cost. This information allows an SoS decision maker to compare designs on the basis of performance and cost as well as risk of not achieving those performance and cost values.

The PR of the component systems can be estimated by the SoS designer, based on approximations for the designer's managerial control over the component and the estimated influence, either monetary or otherwise, that the designer expects to have over the component. In this method, the effective managerial control (and the PR derived from it) is a variable for that particular component. Effective managerial control can be varied over a range generating many SoS designs which can them be plotted on the tradespace.

The PR for a particular SoS design can be derived from the participation risks of the component systems in the SoS. For example, if the PR for component system A is 0.8 and PR for component system B is 0.2, the corresponding 'Likelihood of Participation' (LP) [6] for each is $(1-PR)$, or 0.2 and 0.8 for A and B, respectively. Assuming that the participation of components is independent of each other, the combined Likelihood of Participation for the SoS is the product of the individual components' LP, i.e. 0.2*0.8 = 0.16. Therefore, the combined PR for the SoS design AB would be $(1-LP)$ = 0.84. This PR for each SoS design allows the designer to compare within the tradespace the relative likelihood of achieving diverse SoS configurations. The designer can identify high performance, low cost designs as in traditional tradespace exploration, but the newly added PR component enables the designer to also differentiate between configurations that are easy to achieve versus those that are relatively difficult to arrange. This enables the SoS designer to compare high performance-high risk designs with relatively lower performance-low risk designs that may be a more attractive solution.

5 Steps in the SoS Tradespace Exploration Method

At the earliest stage of SoS concept exploration, the interfaces required between the component systems are not well defined, and thus can be reasonably estimated using the attribute combination complexity factor introduced above, along with a change in cost for the overall SoS system. The focus of the method and the modeling of systems within the method should be towards gaining maximum knowledge about the high-level SoS system trades using the limited time and resources available in early conceptual design. In subsequent stages of the SoS design when the large SoS design space has been reduced to a smaller number of potentially value robust options, the simplifying assumptions can be replaced with more detailed interface design.

The concepts discussed in the above sections can be used to construct a practical SoS tradespace exploration method that can be used by the SoS designer to compare diverse SoS concepts on the same tradespace. The steps from the viewpoint of an SoS designer going from the SoS need identification to the SoS tradespace are as follows:

1. Elicit SoS attributes from the SoS stakeholder set
2. Generate a list of legacy systems that may be used to fulfill the SoS mission (assuming that both legacy and new systems are available)
3. Make list of component system attributes in each Class (0,1,2,3,4). This will require some creativity on the part of the SoS designer, and perhaps input from the component managment. Ross [10] describes the process by which attributes in various classes can be generated using the Class 0 and 1 attributes of the component system. The output of this step is a list of attributes for each component, classified into five categories each.
4. Estimate the Participation Risk for each component system. This is done by estimating the managerial authority and the potential influence that can be exerted by the SoS designer on each component system.
5. For each SoS attribute
   (a) Select the component attributes that can be combined to attain the SoS-level attribute. The selection should be done starting from the Class 0 and Class 1 component attributes (as these will require the least cost), and successively adding on higher class attributes if necessary.
   (b) Find that attribute in the attribute lists for the component systems; select components that may be able to together provide the requisite SoS value in that attribute
   (c) Select the SoS level of attribute combination complexity for the SoS attribute
   (d) Determine the attribute value using the component attributes and a method from within the selected level of attribute combination
   (e) Calculate the single attribute utility for that attribute, using utility information provided by the SoS stakeholders
   (f) If the attribute classes of the component attributes used are beyond 0 or 1, add a proportionate amount to the cost to represent the difficulty of achieving that point in the tradespace
6. Calculate the SoS multi-attribute utility for each design, using the single attribute utilities calculated
7. Plot the utility and cost of each SoS design on a tradespace

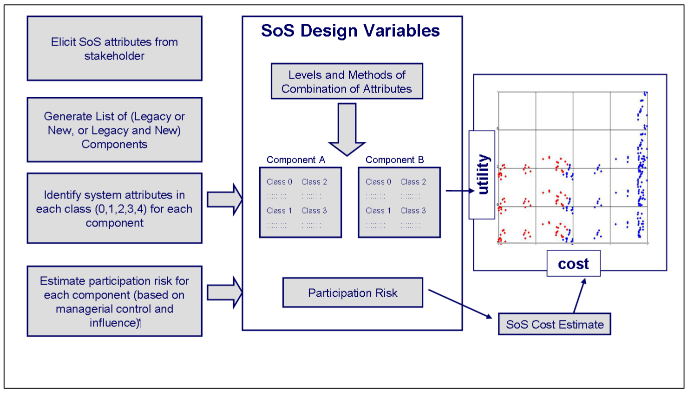
Elicit SoS attributes from stakeholder
Generate List of (Legacy or New, or Legacy and New) Components
Identify system attributes in each class (0,1,2,3,4) for each component
Estimate participation risk for each component (based on managerial control and influence)
SoS Design Variables
Levels and Methods of Combination of Attributes
Component A
Class 0
Class 2
Class 1
Class 3
Component B
Class 0
Class 2
Class 1
Class 3
Participation Risk
utility
cost
SoS Cost Estimate

Figure 3 – *Steps for Modeling SoS Value and Cost*

From the viewpoint of an SoS designer in the early concept exploration phase of SoS design, participation risk and level of attribute combination, as well as the cost to integrate higher class attributes are all estimates. However, varying these parameters in the SoS design space can generate many diverse designs which can then be compared on the SoS tradespace, giving the SoS designer insight into the relative performance and cost of different designs. This allows the designer to identify the key trades in the SoS design, and focus information gathering and design efforts on those aspects. Epoch-Era Analysis [8], a method in which the system timeline is divided into fixed-context snapshots called epochs, enables the designer to compare the performance of these diverse SoS designs over changing contexts. By generating a tradespace for each epoch and analyzing the performance of designs over a wide variety of epochs, the designer can identify value robust SoS designs. Once a possible value robust design has been identified, the SoS designer may employ more detailed methods, such as mechanism design, in order to determine the resources required to obtain that SoS configuration. Thus the method described here provides an initial framework in which a large number of SoS designs can be compared quantitatively, allowing the designer to identify designs for more detailed study. Iterative use of this method with more detailed component system models, and refined knowledge of variables such as participation risk is also possible, providing designers with a tool for even more detailed conceptual design.

6 Application to Example

As an qualitative example of SoS attribute combination, a simple case is considered consisting of two component systems - an aircraft and a satellite. The mission for the SoS is to conduct surveillance. While this is an example case, it is a simplified version of a full-length case study done to design a satellite surveillance system. The attributes that are designated as the SoS attributes are a subset of stakeholder-elicited system attributes for that case study. While this paper is limited to qualitative descriptions of the steps in attribute combination, the method is currently being demonstrated on a quantitative case study and will be published at a later date.

The Class 0 or articulated attributes for the aircraft, satellite and SoS are given in Table 2 .

Class 0 Attributes		
Aircraft	Satellite	SoS
Acquisition Cost	Acquisition Cost	Acquisition Cost
Resolution	Imaging Capability (NIIRS)	Imaging Capability (NIIRS)
Target Track Life	Percentage of Area of Interest Covered	Field of Regard
		Target Track Life

Table 2 – *SoS and Component System Class 0 Attributes*

According to the steps delineated in the previous section, at first, the Class 0 and Class 1 attributes are identified using information about the performance of the system. Class 2 and Class 3 attributes are generated by recombination of Class 0 and 1 attributes, and by enumerating possible modifications to the system design variables, respectively.

Then the SoS attributes are considered one by one, to determine the classes of attributes as well a the level of combination required to achieve that SoS level attribute.

Acquisition Cost Acquisition cost is an attribute for both component systems and the SoS, and thus is a Class 0 attribute for all three. Combining the aircraft acquisition cost attribute and the satellite acquisition cost attribute is a combination of two Class 0 attributes, and thus is relatively easy to accomplish and does not add to the SoS cost significantly. Selecting a low level of combination in this case, a simple combining method of the component attributes such as addition can be used to obtain the SoS attribute value.

Imaging Capability NIIRS levels are a way in which to measure imaging quality [15], and require both imaging resolution as well as additional information about image interpretation. While this is a Class 0 attribute for the satellite, it is a Class 2 attribute for the aircraft (as calculating the NIIRS level for the aircraft requires a combination of the Class 0 attribute of resolution, along with other existing imaging information relating to 'interpretability' [15], to generate NIIRS levels). The cost incurred to combine a Class 2 attribute is approximated at 10 percent of the component system cost, thus this is the additional cost that must be considered for this particular attribute. A high level of combination is selected in this case, since it is a crucial attribute for the surveillance system, and the generation of the SoS attribute is done through data fusion algorithms.

Field of Regard This can be obtained through combination of a Class 1 (latent value) attribute for the aircraft, and a Class 1 (latent value) attribute for the satellite. Thus there is no additional cost for combination considered in this case, as both attributes are already exisiting for the components. The combining method chosen here is medium level, involving simple averaging of the two component attributes to obtain the SoS attribute value.

Target Track Life The continuous length of time that the system can track a particular target is called the target track life. While this is a Class 0 attribute for the aircraft, it is a Class 3 attribute for the satellite, as the primary mission of the satellite is imaging and it would require significant resources to equip the satellite to measure track life. Thus the cost for this Class 3 attribute in the combination is approximated as 30 percent of the satellite component cost, and added to the SoS cost. The combining method chosen is high as it requires tailored algorithms for track life combination. The concept of operation for the SoS plays into this attribute heavily, as there may be a scenario where the satellite identifies a target and passes off the track to the aircraft, resulting in a complex track life calculation.

Once the single attribute values are obtained, the single and multi attribute utility values for the SoS design can be calculated.

Despite the constraints on concept of operation possibilities for component systems in the SoS, determined by the component management, the SoS designer may have several levels of attribute combination complexity to choose from when designing the SoS. To generate more designs in this space using the same two components considered above, the levels and the methods utilized within the levels to combine attributes may be varied. These points can then be plotted on a tradespace to compare SoS designs. Examples of SoS tradespaces generated through the application of aspects of a simplified version of this SoS tradespace exploration method, and the associated tradespace analysis, can be found in [16]. The application of the full SoS tradespace exploration method on a quantitative case study is currently being conducted to demonstrate the method in greater detail.

7 Discussion

The SoS attribute combination method described above is an essential part of a quantitative SoS tradespace exploration method. This method of attribute combination provides a means to quantitatively estimate the SoS value delivery, and thus enables the comparison of diverse SoS designs on a common performance-cost basis. Tradespace analysis methods such as Pareto analysis and Epoch-Era analysis can then be used to identify potential value robust SoS designs. In addition, this method may lead to the identification of undesirable (or possibly desirable) emergent SoS properties through the interaction of component systems. During the second step of attribute combination, many possible attributes of each component system in each attribute class are generated. Through comparison of these lists of attributes, it is possible to identify key interactions between component system attributes that may degrade (or possibly enhance, through the interaction of component systems) the performance of either the SoS or the individual components. As a result, design steps may be taken to prevent undesirable (or encourage desirable) emergent properties from being displayed at the SoS level through use of a system mask, which changes the system as viewed by its context (i.e., masking some of its properties while leaving the component system itself unchanged) [17]. Similarly, from the viewpoint of the local component system stakeholder, if component system value delivery is degraded due to interactions resulting from participation in the SoS, a system shelter construct can be used, which shields the component system from changes in external context. Both system mask and system shelter come at some cost. The SoS designer must take these additional costs and their distribution between the SoS and the component systems into account in any detailed mechanism design.

8 Summary

This paper presents a method by which to model SoS value delivery using component system attributes as well as considerations about the independent management of components, the concept of operation of the components, and the cost of integrating the components into an SoS. The modeling of SoS attributes is a critical aspect of a quantitative SoS tradespace exploration method to be used in SoS early conceptual design, when the design space and parameters are undefined. This paper provides practical steps that can be used to quantify the SoS utility in order to generate SoS tradespaces. It also incorporates the concept of ‘Participation Risk’, which represents the uncertainty of component system participation faced by an SoS designer due to independent component management. Consideration of risks due to the independent management of the component systems is a means by which to approach the complex problem of the multi-level multi-stakeholder SoS value proposition in this method. In the past, the multi-stakeholder problem has been difficult to tackle quantitatively within the system design process. While this method does not solve the problem for SoS, it does provide the SoS designer a framework by which to relatively compare designs that different levels of risk due to the diversity of stakeholder expectations at the local and global level in SoS. The quantitative SoS tradespace exploration method, of which attribute combination is an integral part, aids a SoS decision maker in making design decisions which result in value robust SoS designs.

9 Future Work

This method for combining component attributes into a model for SoS value delivery is currently being incorporated into a full SoS tradespace exploration method that will enable tradespace exploration for SoS as well as identification of value robust SoS designs. This paper presents a simple qualitative example of the application of the method of modeling SoS attributes, but this method is being quantitatively applied to a case study as well. In future work, more detailed study of the types of attribute combination methods available within each level of combination complexity is needed in order to aid SoS designers in practical conceptual design.

10 Acknowledgments

The authors gratefully acknowledge the funding for this research provided by the Systems Engineering Advancement Research Initiative (SEAri), a research initiative within the Engineering Systems Division at the Massachusetts Institute of Technology. SEAri (http://seari.mit.edu) brings together a set of sponsored research projects and a consortium of systems engineering leaders from industry, government, and academia.

References

[1] DeLaurentis, D.A. (2005), “Understanding Transportation as a System-of-Systems Design Problem,” AIAA Aerospace Science Meeting Exhibit.

[2] Dahmann, J.S. & Baldwin, K.J. (2008), “Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering,” in “IEEE International Systems Conference,” Montreal, Canada.

[3] Maier, M.W. (1998), “Architecting Principles for Systems-of-Systems,” *Systems Engineering* 1(4), pp. 267–284.

[4] Keating, C., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Safford, R., Peterson, W. & Rabadi, G. (2003), “System of Systems Engineering,” *Engineering Management Journal* 15(3), pp. 36–45.

[5] Ross, A.M. (2006), *Managing Unarticulated Value: Changeability In Multi-Attribute Tradespace Exploration*, Ph.D. thesis, Massachusetts Institute of Technology.

[6] Chattopadhyay, D., Ross, A.M. & Rhodes, D.H. (2008), “A Framework for Tradespace Exploration of Systems of Systems,” Los Angeles: Conference on Systems Engineering Research.

[7] Ross, A.M. (2003), *Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-centric Framework for Space System Architecture and Design*, Master’s thesis, Massachusetts Institute of Technology.

[8] Ross, A.M. & Rhodes, D.H. (2008), “Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis,” Utrecht, the Netherlands: INCOSE International Symposium.

[9] Ross, A.M., Rhodes, D.H. & Hastings, D.E. (2009), “Using Pareto Trace to Determine System Passive Value Robustness,” Vancouver, Canada: IEEE International Systems Conference.

[10] Ross, A.M. & Rhodes, D.H. (2008), “Using Attribute Classes to Uncover Latent Value During Conceptual

Systems Design," Montreal, Canada: IEEE International Systems Conference.

[11] Richards, M.G., Ross, A.M., Shah, N.B. & Hastings, D.E. (2008), "Metrics for Evaluating Survivability in Dynamic Multi-Attribute Tradespace Exploration," San Diego, CA: AIAA Space.

[12] Hall, D.L. & LLinas, J. (1997), "An Introduction to Multisensor Data Fusion," in "Proceedings of IEEE," IEEE, vol. 85, pp. 6–23.

[13] Steinberg, A. & Bowman, C. (2004), "Rethinking the JDL Data Fusion Levels," Chicago: AIAA Intelligent Systems Conference.

[14] LLinas, J., Bowman, C., Rogova, G., Steinberg, A., Waltz, E. & White, F. (2004), "Revisiting the JDL Data Fusion Model II," in "In P. Svensson and J. Schubert (Eds.), Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004)," pp. 1218–1230.

[15] Assessments, I.R. & Committee, R.S.I. (1996), "Civil NIIRS Reference Guide," http://www.fas.org/irp/imint/niirs_c/guide.htm.

[16] Chattopadhyay, D., Ross, A.M., Borer, N. & Rhodes, D.H. (2009), "Demonstration of System of Systems Multi-Attribute Tradespace Exploration on a Multi-Concept Surveillance Architecture," Conference on Systems Engineering Research.

[17] Ross, A.M. & Rhodes, D.H. (2007), "The System Shell as a Construct for Mitigating the Impact of Changing Contexts by Creating Opportunities for Value Robustness," Honolulu, HI: IEEE Systems Conference.