A CAPABILITY-BASED FRAMEWORK FOR APPLYING VALUE-DRIVEN DESIGN TO SYSTEMS WITH MULTIPLE VALUE-PRODUCING SCENARIOS

A Thesis

by

ROBERT MATTHEW PRICE

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Richard J. Malak
Committee Members,	Mark Avnet
	Daniel A. McAdams
Head of Department,	Andreas A. Polycarpou

August 2016

Major Subject: Mechanical Engineering

ABSTRACT

Value-driven design is an emerging paradigm in systems engineering and design that leverages insights from decision theory to inform decision-making in engineering design and improve systems engineering outcomes. A basic premise of value-driven design is that systems engineers should formalize preferences for system engineering project outcomes using a single overall scalar measure. To date, most value modeling work reported in the literature addresses the value of a system for a single primary usage scenario. The capabilities of the system are typically modeled with the assumptions and needs of this single scenario in mind, with the model of system capability integrated into the value model for this single scenario.

However, many systems support multiple usage scenarios. Optimizing these systems solely for any one scenario may diminish their value on another. Additionally, maintaining multiple models of the system, each tailored for a single value scenario, is effort-intensive and introduces the risk that inconsistencies will develop.

This thesis presents a framework for modeling these types of systems in a way that avoids the pitfalls and reflects a more accurate picture of the true system value. A key aspect of the framework is the clear distinction between modeling the technical capabilities of a system and modeling the value generation of usage scenarios that leverage those capabilities. System capabilities are modeled in such a way as to be compatible with any individual usage scenario while remaining broad enough to address all usage scenarios. This may necessitate expanding the top-level representation of

system capabilities beyond a vector of attribute values to include relationships between attributes. Constructing the system capability model separate from the value models of individual scenarios clarifies the modeling tasks, simplifies model traceability, and grants decision makers flexibility in exploring portfolios of value-producing scenarios.

The framework is demonstrated in a notional example involving the selection of system architectures for a non-commercial space launch vehicle that must support a variety of value-generating missions. In order to illustrate the usefulness of the approach, this example is then expanded upon to investigate the concept of robustness and how it might be valued in such systems.

ACKNOWLEDGEMENTS

I want to thank the NASA Systems Engineering Consortium for funding this work. Any conclusions, opinions, or findings are solely those of the author and do not necessarily reflect the views of NASA.

I want to thank my committee chair, Dr. Malak, for all of the advice and mentoring (both in technical matters and more general life matters) he has given me over the years. I'd also like to thank my committee members, Dr. Avnet and Dr. McAdams, for contributing their time and energy to ensure that this thesis is the best that it can be. I would also like to thank Dr. Jacobs for agreeing to sit in on my defense for Dr. McAdams.

I want to thank my mother and father for raising me to be inquisitive, teaching me communication skills, and always being there for me. Last but most definitely not least, I want to thank my fiancée Katie for her patience, encouragement, and steadfast love. I couldn't have finished this without your support.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

Value-driven design is an emerging paradigm in systems engineering that shows promise as a foundation for decision making in systems engineering. Sometimes also called value-centric design, it involves formalizing one's preferences about system engineering outcomes by constructing a system value model that takes the engineered system's design attributes as inputs and outputs a scalar score indicating system expected value [1]. A system with a higher expected value is a more preferred system. Value-driven design offers guidance to systems engineers in selecting the best overall system design concept available, rather than just the least expensive concept that meets all requirements – which may or may not be the most preferred system overall when all of the decision-maker's preferences are taken into account. Value-driven design also provides guidance and clarity to those using optimization techniques to search the design space because it formalizes a top-level objective function for the entire system. Additionally, some argue that a value-centric approach will avoid pitfalls such as cost and schedule overrun associated with using requirements to specify minimum acceptable levels of key performance attributes [1] or focusing myopically on cost reduction while ignoring the overall value proposition of a system [2].

When developing system value models, value-driven design practitioners often model a system as producing value through some specific scenario, such as being sold for a profit [3], producing a revenue stream over time [4], or performing some specific function of inherent value to the decision-maker [5, 6]. When operating under this assumption of a single value-generating scenario, one can model the system capabilities (the word “capability” in this thesis generally refers to a top-level attribute relevant to system value as opposed to a design variable that may or may not directly impact value) with any assumptions inherent to that scenario integrated into it. Additionally, the needs of the single value-generating scenario determine which attributes of the system are pertinent to value and are therefore included in the modeling effort. The top-level model of system capability is thus inextricably tied to the value model. This approach of modeling system capability according to the demands of a single value model grown out of a specific usage scenario is fairly standard in the literature on value-driven design [7-15].

However, some systems produce value through a variety of substantively different value-generating scenarios. Consider for example a multirole military aircraft designed to perform a variety of mission types ranging from surveillance to destruction of ground targets to aircraft combat. Each of these missions produces value to the decision-maker by making use of different regimes of the system’s capability. It is highly unlikely that the optimal system for one mission will be the optimal system for another mission. Attempting to construct a single value model with the relevant top-level system attributes integrated into it will result in an incomplete picture of this system’s value at best, and maintaining a separate integrated value model for each mission (each with its own customized model of the system) introduces risks that ambiguity and errors will creep into the representation of the system.

Note that there is a distinction between a system that truly has multiple value-producing scenarios and a system which performs a variety of missions that produce different values while utilizing similar capabilities [16], or a system whose operation in a single value-generating scenario produces value for multiple stakeholders [17]. In both of these latter cases, although there is a degree of diversity in value-generating scenarios or in the ways in which the single value-generating scenario produces value, system capabilities can still be integrated into the value model because there are not different scenarios which use substantively different regimes of system capability.

This research develops a novel framework for applying value-driven design to systems that produce value through multiple value-generating scenarios which may be employed in differing amounts and may make use of the system's capabilities in different ways. In contrast to the traditional approach of tying the model of system capabilities to the value model, this work advocates for modeling system capabilities in a general sense and then interfacing that general capability model with a portfolio of value models, one for each distinct value-producing scenario. The phrase "capability model" refers to a collection of all top-level system attributes relevant to system value (whether design variables or system responses), which may include scalars, vectors, and relationships. This is in contrast with the more limited representation of the system used by an approach such as tradespace exploration, which is limited to a multidimensional vector and cannot include relationships or curves. Whether or not a particular system attribute is relevant to system value is determined by the needs of the value models – this is discussed in greater detail throughout Section 3.

This thesis is organized as follows: Section 2 presents background on various paradigms and topics within systems engineering that are of interest to this work. These topics are value-driven design (of which this work aims to be an extension), model-based systems engineering (a philosophy within which this work intends to operate), tradespace exploration (a related paradigm that is nevertheless distinct from this work), and robustness (a desirable, but tricky to measure, attribute of a system whose impact on value can be determined using this method). Section 3 presents the valuation framework in a systematic manner, laying out its various components and a process for implementing it. Section 4 then demonstrates the framework with an example implementation using a publically developed launch vehicle as the system of interest. Section 5 extends this launch vehicle example to show how this framework can be used to answer questions related to robustness and evaluate its impact in a system. Finally, Section 6 presents some closing remarks and summarizes the recommendations of this research.

2. BACKGROUND

2.1. VALUE-DRIVEN DESIGN

Engineers are uniquely equipped to make informed technical decisions, which carries a unique responsibility. The consequences of poor engineering decisions can be disastrous both in terms of economic damage and loss of life. It stands to reason that if there is some normatively correct way to make decisions, engineers should at least be trying to use it. Modern normative thinking about decision-making under uncertainty (the only kind of decision-making with any practical application to engineering as all engineered systems realistically involve some uncertainty) grows out of the theorem of expected utility, first articulated in the seminal work of von Neumann and Morgenstern and later expanded upon by other researchers [18-20]. The concept of engineering as a decision-making discipline began to emerge prominently in the late 80s with Mistree et al. arguing for “decision-based design” as a way of thinking about engineering design [21, 22]. Around the same time, Thurston began advocating for the use of multiattribute utility analysis to evaluate design decisions [23, 24]. More recently, Hazelrigg has repeatedly called for engineers to focus on leveraging normative decision-making methods in systems engineering and design [25, 26]. For a more detailed treatment of decision-making in engineering, see [27].

Value-driven design is a way of thinking about systems engineering that is related to this call to think about engineering as a decision-making discipline, but originating in the systems engineering community (as opposed to the engineering design community). Fundamental to value-driven design is the idea that engineers should strive to design the "best" systems possible, not simply systems that conform to some set of requirements imposed on top-level system attributes. This is accomplished by formalizing evaluation of system design concepts using a scalar measure of value, typically (though not always) based on some economic measure. A system-level objective function or value model is constructed which maps from system attributes to value. In principle, this then allows for the application of optimization methods to determine the best system design as opposed to simply any system design that meets requirements – see Figure 1. In reality, the application of optimization methods to an entire systems engineering project at once may be quite cumbersome, but value-driven design nonetheless offers a clear indication of how to improve a design and can guide decision making in ways requirements (which do not indicate a direction of improvement) cannot.

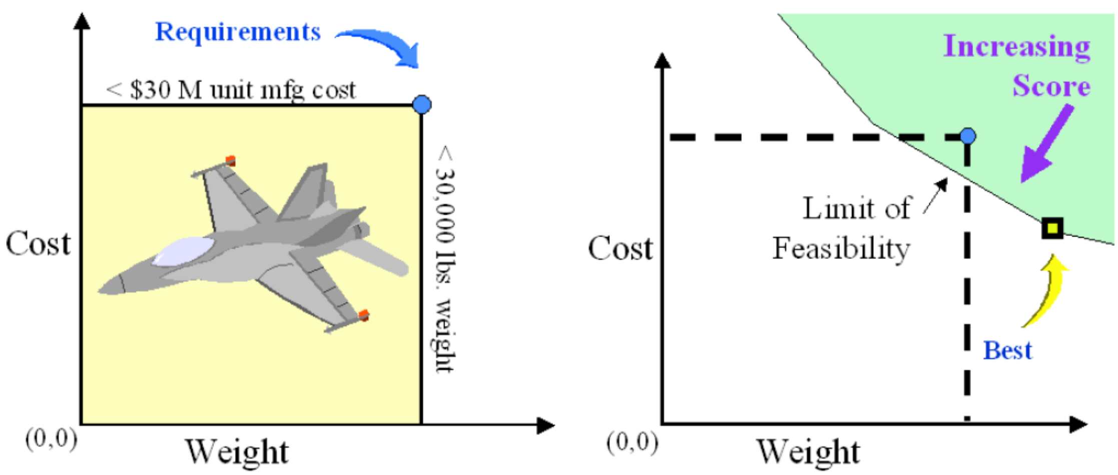


Figure 1. Requirements approach contrasted with VDD approach [1]

The specific structure of this mapping from system attributes to value (or value function) will vary with context, but it generally involves using physics and economic models to relate all the top-level system attributes which contribute in some way to value to the system value through some equation (in the simpler cases – see [7, 13, 17, 28] as well as Section 4.4 of this thesis) or simulation (in the more complex cases – see [3-5, 14, 29-31]).

Once the system-level value function has been constructed, the best system is then the system which maximizes expected value. If desired, a risk attitude can be applied to the output of the value function to incorporate the decision-maker's risk preferences into the model – see Section 5.3 for an example of this. However, for some systems engineering decisions this may not be necessary and expected value is a sufficient final arbiter of the most preferred system – for example, Collopy argued in [32] that in cases of government decisions [33] or more generally, engineering decisions in which the risk is small compared to the decision-maker's risk tolerance, risk attitude can be treated as approximately risk-neutral. For a more detailed treatment of the history of value-centric thinking in systems engineering and design, see Collopy [1]. For further reading on recent developments in value-driven design, see [29, 32, 34-36].

Value-driven design forms the backbone and primary guiding paradigm of the work presented in this thesis. The framework developed in this thesis aims to show how value-driven design can be applied to a specific class of systems (those with multiple value-producing scenarios – see Table 1) more efficiently.

2.2. MODEL-BASED SYSTEMS ENGINEERING

Although the focus of this work is primarily on proposing a framework for applying value-driven design to a specific class of systems, it nevertheless intersects with and is advised by another developing paradigm within systems engineering and design: model-based systems engineering, or (as it is more commonly styled), MBSE [37, 38]. Just as value-driven design might be succinctly summarized by the core directive "design the best system instead of merely one of many acceptable systems", model-based systems engineering might be succinctly summarized by the core directive "formalize knowledge about systems engineering projects, artifacts, and processes using computer-interpretable models with well-defined semantics instead of static documents wherever possible."

A bit of elaboration is in order. Model-based systems engineering is a paradigm in which systems engineers strive to maintain a "single source of truth" (a concept borrowed from information and data management [39-41]) in representations of the system. This eliminates ambiguity, improves traceability, and facilitates collaboration between different groups of engineers. There are a number of practical implications of this. MBSE practitioners take aspects of a systems engineering project that would traditionally have been represented by stacks of documents and instead model them using standard, computer-interpretable representations. This allows changes to be propagated easily, prevents unnecessary duplication of effort, and enforces consistency. Requirements, specifications, operational models, interfaces, test plans – all of these and more are modeled in an unambiguous way within MBSE. This is accomplished through new modeling languages designed to facilitate the practice of MBSE in the systems engineering community.

The most widely used of these modeling languages is SysML [42], an extension of UML [43]. Both of these languages are non-proprietary open standards developed by the non-profit systems engineering consortium Object Management Group. SysML is a visual modeling language that allows for the modeling of many different aspects of systems engineering projects in a traceable way. System behavior, structure, and requirements can all be modeled using SysML [44]. It has been adopted by many in the systems engineering community as a useful language for modeling complex systems. Various authors have proposed extensions or add-ons to SysML that allow for linking it with more traditional executable analysis models [45] or for supporting a value-driven design framework [46], bringing the MBSE ideal – to model as many aspects of the system as possible in a traceable, shareable, and unambiguous way – closer to realization.

Model-based systems engineering informs the structure of the framework presented in this thesis. The impulse to develop a framework in which the system is represented by a single model capable of interfacing with multiple value models (as opposed to giving each value model its own representation of the system) stems from the MBSE guideline to maintain a "single source of truth" when modeling the system.

2.3. TRADESPACE EXPLORATION

We turn now to another paradigm related to, though distinct from, the capability-based value-driven design framework that is the subject of this thesis. Tradespace exploration, sometimes also called tradespace analysis, is a systems engineering paradigm that allows decision makers to explore the set of feasible design solutions using dominance analysis and visualizations to promote intuitive understanding of the set of possible design solutions.

In the tradespace exploration paradigm, feasible system designs are represented as points in an N-dimensional space with N top-level system attributes of interest to the decision-maker forming the dimensions of the space [47]. Each system attribute is assumed to have a monotonic preference direction – either "less is better" or "more is better. [48]" Once the set of feasible designs has been generated, a Pareto frontier can be identified that is the set of all non-dominated points. The concept of Pareto dominance is borrowed from the economics literature [49], and when translated to engineering design it states that a concept is considered dominated if another concept is at least equal to it in every attribute and better than it in at least one attribute. The Pareto frontier is typically taken as the set of optimal solutions, and the decision-maker then chooses a solution from the frontier based on their own intuition and preferences with the aid of visualization tools [50-52].

Tradespace exploration is an extremely useful tool, but has certain limitations. One notable limitation is that traditional tradespace exploration can find it difficult to deal with uncertainty in the system's attributes or its response to uncertain conditions.

Because Pareto dominance entails a comparison of multiple points in a vector space, it is inherently a deterministic concept. It is difficult to address this without sacrificing the rigor of Pareto dominance. Stochastic dominance (see [53, 54]) does allow uncertainty in attribute values to be addressed in a way consistent with decision theory, though it is a cumbersome method to apply. Additionally, Malak and Paredis' parametrized Pareto frontier [55] allows for the accommodation of uncertainty in which attribute values are not currently known but will be when the decision is made. Accommodating uncertainty in tradespace exploration, while possible without sacrificing rigor, is considerably more difficult than in value-driven design due to the assumptions that inherent in dominance analysis.

Another limitation of tradespace exploration is that although it is very useful for increasing decision-makers' understanding of the available design options, it lacks explicit decision-making power because it is not a decision-making method. Tradespace exploration is fundamentally a tool for helping humans to visualize decision scenarios, understand relationships between variables, and make informed decisions. However, without incorporating concepts from value-driven design, tradespace exploration cannot explicitly suggest any decision alternative as the "best." Recent work by Miller et al. has suggested that tradespace exploration visualization tools can be paired with a value-driven design philosophy to work in concert, leveraging the strengths of both approaches to enable better design decisions [56]. In their work, the multi-dimensional visualization tools of tradespace exploration are used to more strongly place the "human in the loop" during the value-driven design process.

One final limitation of tradespace exploration (and that most relevant to this work) is that because it typically represents a system alternative as a point in a multidimensional space, system design information containing relationships or curves cannot be represented. For example, an engine curve relates the torque or power of an engine to its speed. This is a curve that is unique to each engine, and contains a complex picture of the engine's capabilities. However, a tradespace exploration representation of that engine cannot contain the entire curve – only expressions of certain features of it such as the max power output or max torque (and the speeds at which these occur). Some of the information about system capability is lost in this lower fidelity representation. As will be argued, applying value-driven design to complex systems (especially those with multiple value-generating scenarios) may require more nuanced representations of the system than the "point in an N-dimensional space" representation tradespace exploration uses. Further discussion of this can be found in Section 3.6.

Tradespace exploration intersects with this work primarily by contrast – the system capability model presented in Section 3.6 is a more descriptive alternative to the model of the system used in tradespace exploration. This is not to say that this work intends to supersede tradespace exploration – it addresses a different need. Further discussion of the limitations of this work and its links with tradespace exploration can be found at the end of Section 6.1.

2.4. ROBUSTNESS

The final subject to be explored in this background section is not a paradigm as much as a quality (or "ility") of a system – robustness. While nearly any systems engineer would agree that robustness is a desirable quality in a system, ask them to precisely define it and the agreements might end there. Indeed, a precise definition of robustness is difficult to pin down. This is not to say that some authors have not tried. Robustness is an often-discussed subject in both the systems engineering literature and in the technical literature at large. The problem with defining robustness is less that no one has defined it and more that it has too many definitions, with the specific definition often depending on the field [57].

NASA veteran and systems engineer Dr. Mike Griffin identifies robustness as one of the key qualities of an elegant system, but acknowledges that it is not a quality that can be quantifiably measured [58]. In general, robustness is not well-defined nor easy to quantify. Kitano's biological robustness [59] and Taguchi's Robust Design [60] represent two of the most prominent schemes proposed for quantifying robustness. Kitano proposes a scheme based on quantifying the robustness of each individual function of the system. Possible perturbations of the system are identified, and the robustness is calculated from the ratios of function performance under each perturbation to nominal performance. A system function that loses less functionality in response to perturbations is said to be more robust to that perturbation. Taguchi's Robust Design adopts a target-seeking variance minimization scheme in which some system response variable is assumed to be optimal when at a certain target value, and any deviations from that target are assumed to degrade system quality, usually through a quadratic loss function. A more robust system is one whose response variance is minimized.

Recently, Malak et al. examined robustness specifically from a normative decision-making perspective [61] – the same perspective, it should be noted, that underpins value-driven design. They concluded that attempts to quantify robustness directly such as those just discussed tend to be problematic from a normative decision-making perspective for a number of reasons. Instead, they argue, systems engineers should take robustness into account by modeling the beneficial effects of what would qualitatively be called "robustness" in their value models, ensuring that they take into account all the relevant operating scenarios, and using a utility function to express a risk preference if desired. Proper application of engineering methods grounded in normative decision theory will, they argue, result in systems engineers designing the systems they want. If robustness is desired, it will thus be taken into account and valued without being explicitly quantified.

This work adopts the perspective on robustness developed by Malak et al. [61]. This is not an arbitrary choice. Their perspective, one might argue, is a natural fit for any value-driven design method as the two share a common presupposition: the recognition that normative decision theory should be used as the foundation for engineering decision-making. Robustness quantification scores or schemes may be useful for informing concept generation, inspiring new solutions, or granting insight into the system – but not as direct figures of merit used as inputs to a value model. Instead, value models should be constructed in such a way as to model the benefits of attributes qualitatively associated with robustness and other related qualities such as reliability or flexibility. Section 5 will present a few examples exploring how this valuation framework might be used to assess the benefits of robustness in this way.

3. A CAPABILITY-BASED VALUATION FRAMEWORK

3.1. SINGLE VS MULTIPLE VALUE SCENARIOS

Engineered systems are often designed to produce value for the decision-maker through some specific value-producing scenario, such as being sold for a profit or accomplishing some goal of inherent value to the decision-maker (such as destroying military targets or performing scientific experiments). When this is the case, that scenario provides the context for the system value model. This then dictates how system capabilities (recall that “capability” in this thesis refers to a top-level attribute relevant to system value as opposed to a design variable that may or may not directly impact value) are modeled because the only attributes which are important for a value model are those salient to the primary value-producing scenario of the system. The value model (stating the mapping between the system’s attributes and value produced) and the system model (encapsulating the attributes of the system relevant to value) are in this case integrated together.

Consider as an example an aircraft designed by a for-profit firm to be sold to a commercial transportation companies. From the point of view of the firm, the system produces value by selling to their customers for a higher price than the cost of development and manufacture. The value-producing scenario for this system is that of being sold to commercial transportation companies for a profit. Therefore, provided that the customers’ demand for the vehicle can be modeled as a function of vehicle attributes, the firm can use net present value of profit as the measure of value. It is important to note here that while individual engineers may care intrinsically about performance for the sake of performance, from the point of view of the firm system performance and other attributes only matter insofar as they make the system more attractive to the customers or less costly to produce. Also note that for a firm with a goal other than profit (such as a governmental agency concerned with serving the public good), the appropriate measure of value may be different, though can often still be represented as a monetary measure. The value-producing scenario drives the importance of attributes.

For this system in this context, system attributes will be considered only within their regime of relevance to the value-producing scenario – for example, in the case of structural integrity, load cases considered for this system will probably not include small arms fire or anti-aircraft weapons, just environmental loads and possibly incidental impacts (such as birds). Additionally, attributes such as the ability to avoid radar detection are not important since the intended customers have no use for them and they will not increase demand for the product. In fact, because attributes such as this do not substantially affect net present value of profit within the specific context of the system's use they will probably not even be included within the model of system capability. Only those attributes or attribute regimes which are relevant to the value-producing scenario are taken into account.

Consider what happens, however, when the "single specific value-producing scenario" criterion does not apply. What if the firm desires to expand to a new market segment and is thus designing the aircraft to appeal to both commercial air transportation providers and governmental agencies? Suddenly, attributes which may have only been important within some restricted regime or may not even have been included in the original system model are now important to properly determining the system's value. The aircraft may now benefit from resistance to small arms fire or from the ability to avoid radar detection. The value model and integrated model of capabilities, previously sufficient, is now no longer a complete picture of system value. Furthermore, it is not inaccurate to say that in a very real sense the system now has multiple avenues of producing value that place different demands on different attributes of the system. The contribution of each of these to total system value must be considered.

This example illustrates a problem systems engineers can encounter when trying to apply value-driven design to systems with multiple avenues of value generation. The need to consider multiple value-producing scenarios complicates the construction of the system model significantly. In these cases, the standard approach of integrating the model of system capability into the system value model is no longer appropriate. Note that the example used in this case was a profit-driven system, but this situation can also arise when dealing with systems that produce intrinsic value to the decision-maker, such as the detailed public launch vehicle example presented in the next section. Table 1 summarizes some common characteristics of systems with multiple value-producing scenarios.

Table 1. Characteristics of systems ill-fitted for a traditional value model

Characteristic	Description
Multiple value scenarios	The system will be used in a variety of distinct and independent value-producing scenarios.
Varied capability utilization	Each value scenario may utilize different aspects or regimes of system capability to produce value.
Some capability irrelevance	An aspect of system capability vital to one value scenario may be irrelevant to another.
Varied scenario frequency	Value scenarios may vary widely in frequency, with some common and others comparatively rare.

Dealing with these sorts of systems demands an approach which affords the systems engineer flexibility in incorporating the multiple value producing scenarios into the overall model, but without setting aside the vital link between the model of system capabilities and each value-producing scenario. After all (as previously mentioned), the model of system capabilities must grow out of the needs of the value-producing scenario(s) in order to properly reflect the link between system attributes and value produced. How are systems engineers to evaluate these systems within a value-driven design context?

One approach would be to construct a traditional value model for each value-producing scenario separately, integrating into each scenario value model a tailored system model containing only the aspects of system capability that are relevant to it. This could be thought of as the natural extension of current practice. The basic structure of such an approach is shown in Figure 2.

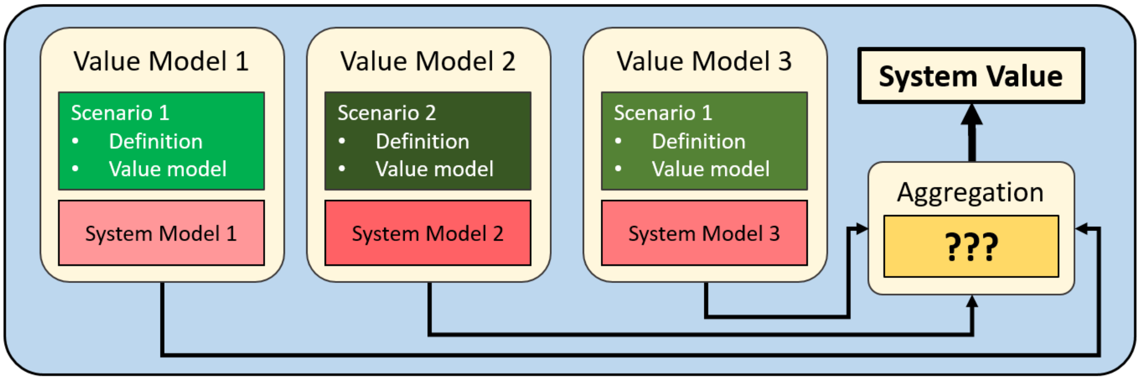


Figure 2. Basic structure of multiple value model approach

Note that this figure uses the phrase "system model" to describe the models of system capability incorporated into individual value models because they will generally not conform to the general definition of "capability model" used throughout this work – the phrase "system model" is thus used to contrast current practice with the approach suggested in this thesis. The definition of "capability model" used throughout this thesis is, to reiterate, "a collection of all top-level system attributes relevant to system value (whether design variables or system responses), which may include scalars, vectors, and relationships." Whether or not a particular system attribute is relevant to system value (and thus included in the capability model) is determined by the needs of the value models that map from system attributes to value; see Section 3.5. In contrast, the models depicted in Figure 2 may vary in their content – typically analysis models will be included in order to transform design variables into system responses. These analysis models are important, but in the parlance of this thesis they are used to help construct the system capability model and in some cases to simulate value production (see Section 3.8) – they are not strictly speaking a part of the system capability model.

The approach shown in Figure 2 is problematic for a number of reasons. The first of these is that it is inefficient to maintain a separate system model for every single value scenario. Propagating design changes through a multitude of slightly different system models is time-consuming, creating opportunities for ambiguity or errors. Maintaining multiple (possibly contradictory) “sources of truth” about a single system is not advised, especially from the perspective of a model-based systems engineering mindset (recall Section 2.2). Additionally, this approach offers little guidance as to how to aggregate or weight the outputs of these disparate value models, which can be problematic if the various value scenarios are utilized with differing frequencies. A major strength of value-driven design (some might say the entire point of a value-driven design approach) is the ability to obtain a single scalar score for each system concept that can then be used to rank-order decision alternatives in a traceable and unambiguous way. A methodology whose output is multiple scores that must then be traded off against one another is of less use.

In contrast to the more cumbersome approach shown in Figure 2, this thesis presents an approach that takes the variety of value-producing scenarios into account and models both system capability and value generation accordingly. In this proposed framework, each value-producing scenario has its own value model. Each of these value models is an independent entity, accepting as inputs whatever system attributes are necessary to determine value generation for that scenario. However, in contrast to the previous approach which integrated a separate system model into each individual value model, in this approach a single general system capability model is developed to be compatible with any value scenario. This capability model is interfaced with each of the value models in whichever way is appropriate and total system value generation is computed. The basic structure of this capability-based framework is shown below in Figure 3.

It should be noted that while the following sections elaborate upon the structure of the valuation framework, the details of implementation will not always be strictly prescribed. This is intentional – the nature of systems engineering projects demands that any general framework for systems engineering and design allow some freedom in the specific implementation of the framework to accommodate the specific context of the project. Nevertheless, a general roadmap for implementation, as well as some general guidelines and best practices, will be given to aid the interested reader in adapting this framework for their own purposes. Figure 4 shows the basic steps to implementing this framework.

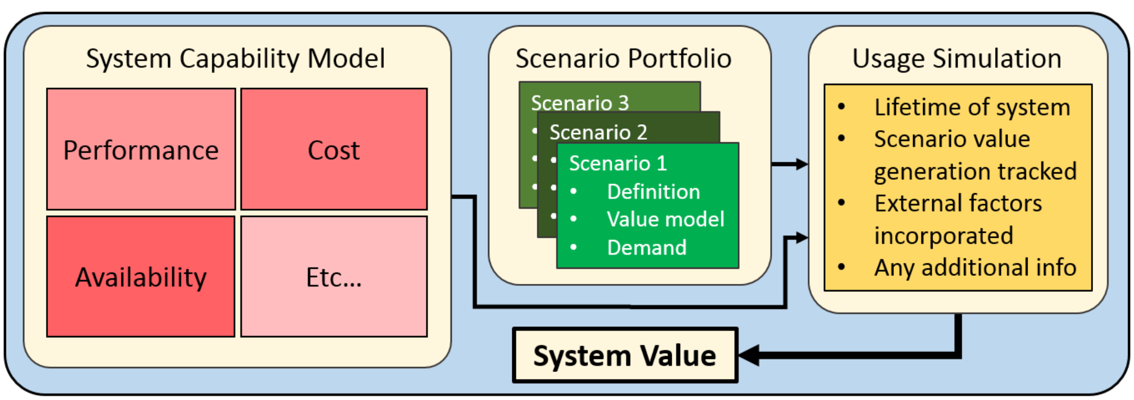


Figure 3. Basic structure of capability-based value framework

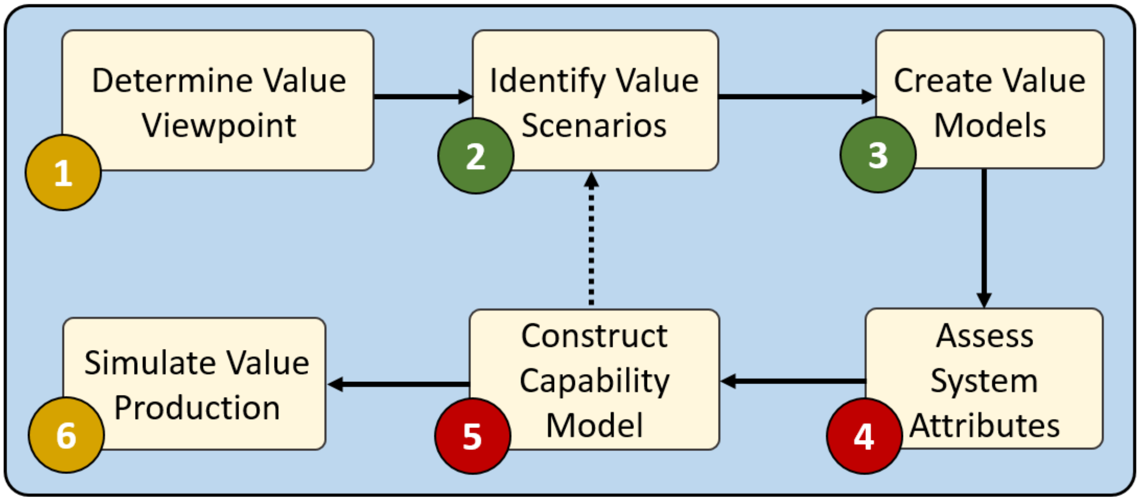


Figure 4. Steps to implementing the framework

3.2. STEP 1: DETERMINE VALUE VIEWPOINT

The first step in implementing this framework is to determine whose value should be considered. Or to put it another way: given that the value-driven design approach says systems engineers should design the "best" system, the first step in doing so is to answer the question "best to whom?" and subsequently the follow-up question "what does 'best' mean for this decision-maker?" Note that the need to answer these questions is not unique to this framework, but a basic prerequisite to doing value-driven design. All value models must be constructed with a specific viewpoint in mind, both regarding whose value is considered and how that value is measured.

Regarding the first question: in general the viewpoint of importance is that of the firm conducting the project. Note that this is one of the differences between "pure" decision theory and value-driven design: decision theory assumes a single decision-maker, while any decision framework used for systems engineering and design must practically be able to accommodate an organization as the “decision-maker.” After the value viewpoint is established, the second question must be answered: how is value to be measured? As mentioned in the previous section, for a profit-driven firm the answer can be fairly straightforward – net present value of profit is often a reasonable surrogate for value. Granted, this reckoning of net present value of profit may be complicated – in some cases firms trade near-term profit for market share, or accept long-run losses on one project because it considerably benefits another. In these cases, the firm may need to consider a larger scope for determining system value such that the effect of the system on net present value of profit for the firm is properly evaluated. For a non-profit-driven firm (such as a public agency), the meaning of value can be even murkier. In some cases, methods such as contingent valuation [62] can be used to “price out” public goods and services. In other cases, scientific value or some other measure of value may be used. The variety of methods for valuing public achievement are beyond the scope of this work – it is assumed that the firm has a way of valuing its goals.

3.3. STEP 2: IDENTIFY VALUE SCENARIOS

The next step in implementing the framework is to develop a portfolio of value-producing scenarios with corresponding value models. This portfolio should be developed systematically. First, all unique value-producing scenarios should be identified. Deciding what qualifies as a unique scenario for the purposes of this framework may require some judgment. A good rule of thumb is: if two modes of operation, use cases, missions, or other similar divisions in system functionality utilize different regimes of system capability or have substantially different relationships between system attributes and value produced, they should be classified as separate value-producing scenarios. There may be multiple useful ways to construct a portfolio of value scenarios. As with many modeling tasks, there is not necessarily a single “correct” model – just models of varying degrees of usefulness (some of which are, incidentally, useless).

3.4. STEP 3: CREATE VALUE MODELS

Once the value-producing scenarios have been identified, they should then be explicitly defined and modeled. Two pieces of information are needed at a minimum to define each scenario:

- An independent value model or value function capable of mapping from system attributes to value
- Information about demand for the scenario or how often the scenario will be utilized

Each unique scenario requires a value model specific to that scenario. These value models are not substantially different from traditional value models. They take system attributes as inputs and output a scalar measure of value. Note, however, that although each scenario value model will use attributes of the system as inputs, there is no requirement that they all use the same attributes. In fact, the nature of the multi-scenario value framework suggests that they will not do so. It is precisely because of this that a general capability model, developed independent of any single scenario but capable of serving all scenarios, is necessary. Each of these value models may take the form of a simple equation (as in the example shown in Section 4), or it may be a more complicated simulation of some sort – see Section 2.1 for references to various examples of both of these. The level of detail is entirely up to the discretion of the systems engineer – what is required is some sort of "black box" that can take system attributes as inputs and output a scalar value.

The other major piece of information needed to model each scenario is information about demand for each scenario, or how often each scenario will be utilized. The multi-scenario value framework allows considerable latitude as to how specific this information must be. It could be expressed in terms as simple as "50% of the time, system will operate under scenario A" or as complex as "no more than once per 2 years, system will have a 15% chance of operating under scenario B for a period of one month". The representation of demand should be as complex as is necessary to accurately portray the relative frequency or weight of each scenario in such a way that the overall usage of the system can be accurately simulated.

3.5. STEP 4: ASSESS SYSTEM ATTRIBUTES

After the portfolio of system value scenarios has been developed, the next step is to identify which top-level system attributes are needed to evaluate the value produced by the system in any of these value scenarios. The model of system capabilities will need to be capable of interfacing with any and all of the system value models. Attributes identified in this step may fall into one of three main categories:

- Attributes required as direct inputs to a scenario value model
- Attributes required as implicit inputs to a scenario value model
- Attributes required in order to simulate overall system value, but not required specifically for any value model

Attributes required as direct inputs to a scenario value model are just that – system attributes which one or more value models require directly in order to calculate or simulate value. Attributes required as implicit inputs to a scenario value model are attributes which may not be direct inputs to a value model, but are still required to calculate or simulate value. One example would be reliability – if a value model specifies that value is only produced if the system does not fail, the chance of failure is implicitly required in order to determine value. Attributes which are required in order to simulate value but not for any value model will vary – in general, these are attributes that are related to the overall usage of the system or the interfaces between value scenarios. One example would be the time in-between missions (due to maintenance, repairs, etc.) for a system which produces value by performing missions. See Section 4.5 and Table 2 for a breakdown of the attributes that fall into each of these classes in the launch vehicle example system.

3.6. STEP 5: CONSTRUCT CAPABILITY MODEL

After assessing which system attributes are needed to interface with the portfolio of value models, the next step is to create the system capability model, which is a comprehensive collection of all top-level system attributes relevant to system value. This may include both design variables and system responses, and may include scalars, vectors, and relationships. This capability model is quite similar to the system model used in the traditional approach to value-driven design, but with a few additional considerations taken into account. Because the system produces value through a variety of different and distinct scenarios, the capability model must be broad enough to be used as an input to all value-producing scenarios. Due to this consideration, the capability model for the complete value framework may actually look very different from the system attributes model one would construct for any single value scenario alone.

The exact form of this capability model will vary with context. The simplest mathematical form of the capability model would be a multidimensional vector of scalar system attributes. This representation could be extended by allowing the individual attributes to be scalars or discrete options, which would be a similar system representation to that used in tradespace exploration. However, due to the nature of these problems a still more complex representation of the system can be necessary. Often certain aspects of capability that could be expressed as scalar values or vectors for any single value scenario must be represented as curves or envelopes in the general capability model because the existence of multiple value scenarios necessitates the additional information encapsulated by the relationships. One example of this need for a more complex system representation is demonstrated by the “delta-V vs payload mass” curve in the launch vehicle example of section 4, which would not be needed in its entirety for any single mission value model but must be included in the general capability model for the capability model to be compatible with any general mission.

The general capability model representation advocated for this approach is thus more complex than the representation typically used in tradespace exploration. This contrast is shown in Figure 5.

The capability model should contain a complete picture of system capability that contains all information about the system necessary to evaluate the value it produces when operating under any individual value scenario.

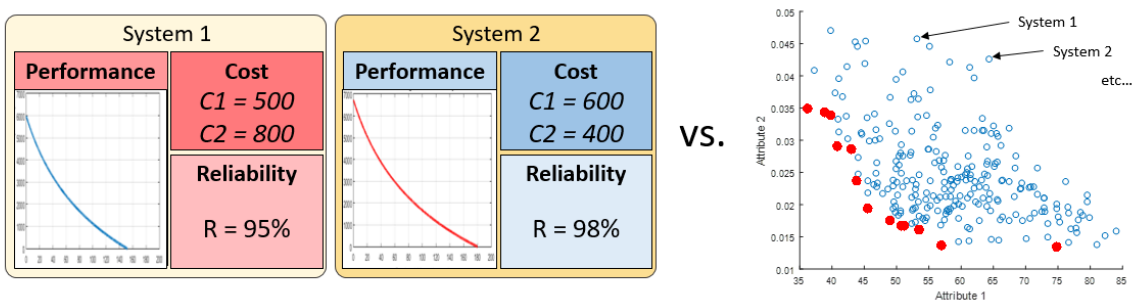


Figure 5. General capability model contrasted with tradespace exploration representation

3.7. STEP 5.5: ADD NEW VALUE SCENARIOS

As shown in Figure 4 by the dotted line between steps 5 and 2, sometimes the process of implementing this framework will operate in a cyclical fashion. Systems engineering projects can undergo significant changes in scope throughout the development process, and this framework is intentionally constructed with the ability to accommodate that contingency. If additional value scenarios arise at any point during the development process, they should be added to the value scenario portfolio. This in turn may prompt a revision of the capability model, as well as of the system itself, to include new attributes which were not previously included in order to accommodate the new value scenario. The ability to work in such a cyclical manner is one advantage of this framework over an approach that develops a separate system model for each value scenario. If the system changes in response to new operating conditions, only the single capability model needs to be changed. This consolidation of system capability modeling reduces opportunities for inconsistency to develop between multiple models and reduces the amount of effort needed to accommodate such changes.

3.8. STEP 6: SIMULATE VALUE PRODUCTION

After the capability model and the scenario portfolio have been developed, the value of the system is evaluated by simulating its usage across some time horizon of interest considering all relevant value-producing scenarios. Information about relative scenario utilization frequency should be used to structure the simulation, and any external or environmental factors affecting the operation of the system should be included in the simulation in whatever detail is necessary to capture their impact on value generation. The capability model is used as an input to all individual scenario value models, and both costs incurred and value generated through all value scenarios are tracked and (if relevant) adjusted for time-discounting. If the capability model, value models, external factors, or any interaction between them is stochastic, an appropriate method should be used to estimate the distribution of system value production and determine the expected value of the system.

4. APPLICATION OF FRAMEWORK: LAUNCH SYSTEM EXAMPLE

4.1. INTRODUCTION

The previous section outlined the capability-driven value framework in very general terms. This section will provide some concreteness to that outline by presenting a notional example problem in which the capability-based value framework is applied to evaluate system design alternatives within a decision context ill-suited for a more traditional value-driven design approach that assumes a single value-producing scenario. The system of interest in this example is a publically developed space launch vehicle. Beyond providing a demonstration of how a real-world system might be modeled and evaluated using this framework, this launch vehicle example will also be used to illustrate how the framework allows for the evaluation of the differences in value associated specifically with robustness in a way that would not be feasible without separating the system model and mission value models.

4.2. PROBLEM SETUP

Suppose that a public space agency, such as NASA, is designing a new launch vehicle. This launch vehicle is intended to serve as the agency's primary all-purpose launch vehicle for at least the next 30 years, and will need to be capable of being utilized for a wide variety of missions – near-earth and solar system exploration, scientific and military, manned and unmanned. The various missions may require different regimes of the launch vehicle's capabilities. For example, a mission to place a commercial satellite in low earth orbit requires significantly less energy from the launch vehicle (and therefore can accommodate a much larger payload) than a mission to send a manned capsule to Mars. Some missions may be conducted as often as there are available vehicles to enable them, while others (due to planetary positional constraints or programmatic restrictions) may only be conducted a few times throughout the entire lifetime of the vehicle program. This variety in mission frequency and utilization of vehicle capabilities makes the capability-based framework ideal for evaluating this system.

Note that the data for both the launch vehicle design alternatives and mission value models is not based on specific real-world vehicles or mission architectures, but is for illustrative purposes only. Attempts have been made to keep the numbers reasonable and within orders of magnitude of what would be expected, and justification for modeling decisions will be presented where appropriate. Nevertheless, note that the primary focus of this example is to demonstrate how to apply the capability-based framework for value-driven design, not to dictate best practices specifically for modeling launch vehicles and space missions.

4.3. STEP 1: DETERMINE VALUE VIEWPOINT

As mentioned in section 3.2, the first step of implementing this framework is to determine whose value is being considered and how it will be measured. This example will follow the advice of that section and simply consider the firm conducting the project (the public space agency) to be the viewpoint of importance. Determining how value will be measured is a less trivial task, however, as a public space agency cannot simply use net present value of profit as a surrogate for value because it is not a profit-driven firm. Rather, it is driven by numerous competing (and sometimes contradictory) goals and strives to please numerous stakeholders. What is more, the relationships between the various stakeholders, their needs, and the outputs of the space agency can manifest as extremely complex networks of value wherein "products" for one stakeholder may be "inputs" for another [63].

Recognizing the existence of these complex interactions between the various stakeholders in public space exploration and the various products (both tangible and intangible) a public space agency produces, we will nevertheless in this example simplify this relationship by assuming that the launch vehicle itself (the system of interest) produces value primarily through enabling missions. Furthermore, we will assume that any individual mission has benefits – whether scientific, military, economic, or other – that can be "priced out" in dollars, allowing all costs and benefits to be compared directly in order to yield net value. Granted, it can be argued that this is a difficult assumption to make. One need only look at the literature on one of the most prominent methods for placing a dollar value on public goods, contingent valuation, to see that the question of how exactly to do this is far from a settled topic [62, 64-68]. However, this is a necessary assumption if we are to be doing value-driven design. Value-driven design, and the normative decision theory which is its foundation, holds that the all of the decision maker's benefits and disbenefits can in principle – and should in practice wherever possible – be expressed in the same units, whether those are units of utility or dollars. The question of how exactly to quantify the value of intangibles such as scientific value is beyond the scope of this work. Normative decision theory suggests that such a thing is possible and it will thus be assumed for the sake of this example.

4.4. STEPS 2-3: IDENTIFY VALUE SCENARIOS, CREATE VALUE MODELS

While a public space agency would in reality likely need to deal with mission types numbering closer to ten or twenty, for the sake of brevity we will consider four representative missions: a mission carrying a generic satellite to low earth orbit, a mission carrying a generic satellite to geostationary orbit, a manned lunar mining mission, and a manned Mars exploration mission. As per the structure of the capability-based value framework, each mission has its own individual value model. These value models are simplified examples of what space mission value models might look like in practice, with numbers that are reasonable within an order of magnitude – they are not derived from any specific real-world value models. Real-world space mission value models would be considerably more complex (see Keller and Collopy [14] for one example) and would need to be developed with input from subject matter experts.

The mission value models are characterized primarily by the following attributes: the amount of delta-V (a normalized measure of impulse imparted to the payload) required to perform the mission assuming a circular orbit at 200 km altitude/28° inclination as a starting point, the cost of attempting the mission, the cost of a catastrophic mission failure, the demand for the mission as a percentage of total missions, and the nominal average value produced per kg of payload.

This nominal value per kg is then multiplied by factors calculated from the injection accuracy, internal fairing volume, and the number of days the launch is delayed. These are abstractions meant to model the changes in mission value brought about by the related changes in vehicle capability. The fairing volume and injection accuracy factors represent the additional value from allowing more space for bulkier cargo and allowing the payload to spend less station-keeping/maneuvering fuel on course corrections, respectively. The delay factor represents the loss of value due to rendezvous or trajectory issues from failing to launch on time. Note that the delay factor is distinct from the cost directly associated with having to scrub a launch (which is modeled as 25% of the nominal launch cost and is not directly tied to any mission value model).

Each mission has distinct values for each of these attributes. The value produced in any given successful run of a mission is then calculated according to Equation (1):

$$Mission\,Value = k_A \cdot k_V \cdot k_D \cdot B \cdot m_p(\Delta V) \tag{1}$$

B is the nominal mission value per kg, and k_i are the factors for accuracy, volume, and delay. $m_p(\Delta V)$ is the payload mass delivered given the required delta-V of the mission, and is interpolated from the curves calculated by Equation (2) and shown in Figure 7, which are both found in section 4.5. The portfolio of mission value models is shown in Figure 6.

	Sat to LEO	Sat to GEO	Lunar Mining	Mars Exploration
Delta-V Required past 28° LEO (m/s), ΔV:	0	4300	4000	5000
Mission Cost ($M):	$100	$125	$200	$750
Failure Cost ($M):	$50	$50	$1250	$2000
Value per kg ($), B:	$12500	$20000	$25000	$80000
Accuracy Factor, k_A:	$1-0.2*A^2$	$1-0.2*A^2$	$1-0.3*A^2$	$1-0.5*A^2$
Volume Factor, k_V:	1+0.00007 *(V-2000)	1+0.00007 *(V-2000)	1+0.0001 *(V-2000)	1+0.00012 *(V-2000)
Delay Factor, k_D:	$0.95^{(\text{Days Delayed})}$	$0.95^{(\text{Days Delayed})}$	$0.9^{(\text{Days Delayed})}$	$0.8^{(\text{Days Delayed})}$
Demand:	45%	35%	15%	5%

Figure 6. Mission value model portfolio

Before moving on to the construction of the system capability model, a brief discussion of a few of the modeling choices made in this section will be presented in order to contrast those that are intended as prescriptive with those that are simply made for simplicity.

First, it is acknowledged that relating value to payload mass may not seem like the most obvious way to “price out” the mission value – after all, not every instrument on a satellite is equally valuable. However, this approach is necessary because the value model must map from attributes of the system to value. The specific value of any given kg of payload is not an attribute of the system, but the mission. The total payload mass available for any given mission delta-V, on the other hand, is (essentially) an attribute of the system. Improving the system design cannot make the mission payload more inherently valuable, it can only increase the total payload mass available any given mission. Therefore, in order to allow the mission value models to map from system attributes to value, mission value must be in some way related to payload mass. For simplicity, a linear relationship between the two is assumed in this notional example – though in a real-world situation the form of the relationship could vary mission to mission and would likely based on detailed estimates specific to the mission. Thus, while the linear form of the relationship between payload mass and value in this example should be taken as a simplification, such a relationship should still exist in some form in a more complex real-world application of this framework to evaluating launch vehicles as it grows out of the basic tenets of value-driven design (relate system attributes to value).

On the other hand, the use of multiplicative factors to model the added value from injection accuracy and fairing volume as well as the lost value from launch delays is acknowledged to be a simplification. In the absence of literature containing mission value models that address these attributes directly, the functional forms of each were chosen to be reasonable approximations of how attribute each might affect the overall mission value. This said, there is little reason to believe that the calculation of overall mission value for a real space mission would necessarily take a multiplicative form such as that shown in Equation (1). This form, as one of the simplest of the possible representations, is chosen in order to take these factors into account for this notional example. In a higher fidelity representation, the value impact of payload mass, injection accuracy, fairing volume, launch delays, and many other attributes could be related in much more complex ways than simple overall multiplicative factors – for example, some attributes could function as complements (more valuable if both are present) or substitutes (less valuable if both are present). The question of how best to relate these attributes (and others) to total mission value could very well vary with each mission, and is a matter for further investigation. Elicitation techniques used in multiattribute utility theory may prove useful in developing these relationships for specific missions [20].

4.5. STEPS 4-5: ASSESS SYSTEM ATTRIBUTES, CONSTRUCT CAPABILITY MODEL

With the portfolio of mission value models developed, the next step is to construct capability models for each candidate system concept. This example will assume that the design of the launch vehicle has been narrowed down to two major concepts which are similar in overall structure and configuration but different in certain key attributes that affect value delivery in various ways. Data for these launch vehicle concepts is reasonable for a general purpose heavy lift launch vehicle, but is not directly based on any specific real-world vehicle designs.

Before constructing the system capability models, it is necessary to assess which system attributes even need to be modeled. Recall that only system attributes with any impact on value need to be included in the capability model. The portfolio of mission value models thus dictates which system attributes should be included in the capability model. Looking at the value models, three distinct classes of system attribute can be identified: attributes required explicitly as direct inputs to a mission value model, attributes required implicitly in order to calculate value, and attributes required to simulate the total system lifetime. Each of these will be addressed in turn.

First we consider system attributes required explicitly as inputs to a mission value model. Referring back to Equation (1) and Figure 6, it is apparent that these attributes are: the payload mass for each mission (as determined by the mission delta-V), the injection accuracy, the internal fairing volume, and the total cost for each launch of a rocket (here represented by the sum of the vehicle manufacture cost and the launch cost).

Next we consider system attributes required implicitly in order to calculate value: system reliability (chance of launch failure) and chance of weather delay. These attributes are not directly used to calculate value in the sense that they appear in Equation (1) or Figure 6, but they are necessary in order to determine for any individual launch (a) if it succeeds or fails and (b) how many days the launch is delayed. Note that although both of these attributes are treated in this example as if they are solely properties of the system, this is not precisely the case. Both system reliability and chance of weather delay also depend on the launch location because they are affected by wind speeds, ambient temperatures, and other environmental factors. A higher fidelity model might specify the conditions under which the launch would fail or under which it would be scrubbed (both properties of the system) and then determine from simulations of the environment at the launch site how likely these conditions are to occur, yielding probabilities of failure and weather delay respectively.

Finally we consider system attributes not required for any individual mission value model, but required in order to simulate the total system lifetime: the development cost of the vehicle and the production/rollout time for each vehicle. The development cost is required in order to determine the initial cost for the system concept before it starts generating value (it is of no use if it never generates more value than the development cost), and the production/rollout time is required in order to determine the interval between each mission when the lifetime of the vehicle is simulated day-to-day. The program is assumed to only have a single launchpad, so this time drives the interval between missions. A summary of the attributes discussed is shown below in Table 2.

Table 2. Attributes required in system capability model

Why needed	Attributes
Explicit inputs to value models	Payload Mass as a function of Delta-V Injection Accuracy Fairing Volume Total Cost per Launch
Implicitly required to evaluate value models	System Reliability Chance of Weather Delay
Required for simulation	Development Cost Production/Rollout Time

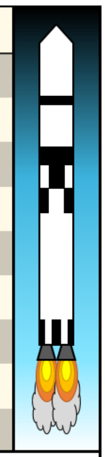
Note also that although most of these attributes are simply scalars (and could thus be represented in an N-dimensional vector), the payload mass for each mission must actually be represented in the capability models by the complete relationship between delta-V and payload mass. This is necessary because the payload mass is a required attribute for each mission, but each mission utilizes a different regime of vehicle capability by requiring a different delta-V. Simply stating the payload mass to some reference orbit as the measure of vehicle payload capacity, as is common practice for more limited launch vehicles designed to serve only a few specific mission types, does not fully encapsulate the necessary information for properly evaluating the concept when multiple diverse missions are involved in the production of value. Such a limited representation would also make the introduction of new mission value models more cumbersome as the required data for interfacing with these value models would not exist in the capability model.

This delta-V vs payload mass relationship is established using the ideal rocket equation [69] for a 2-stage rocket, shown below in Equation (2). I_{spi}, m_{si}, and m_{fi} are the specific impulse, rocket mass, and fuel mass respectively of stage i. g_0 is the standard gravitational constant, and m_p is the payload mass.

$$\Delta V = I_{sp1} \cdot g_0 \cdot \ln \frac{m_{s1} + m_{s2} + m_{f1} + m_{f2} + m_p}{m_{s1} + m_{s2} + m_{f2} + m_p} + I_{sp2} \cdot g_0 \cdot \ln \frac{m_{s2} + m_{f2} + m_p}{m_{s2} + m_p} \tag{2}$$

The capability models for the two concepts are presented in Figure 7, followed by a general description of each of the two design concepts.

	Concept 1, Reliable Mainstay Rocket	Concept 2, Heavy Experimental Rocket
Engine Specific Impulse, I_{sp} (s):	{330, 465}	{350, 485}
Development Cost ($M):	$10000	$12000
Cost per Vehicle ($M):	$1000	$1200
Launch Cost ($M):	$100	$120
Production/Rollout Time (days):	90	105
Reliability:	98%	97%
Chance of Weather Delay:	7.5%	15%
Payload Volume (m³), V:	3500	2000
Injection Accuracy (km), A:	0.5	0.1



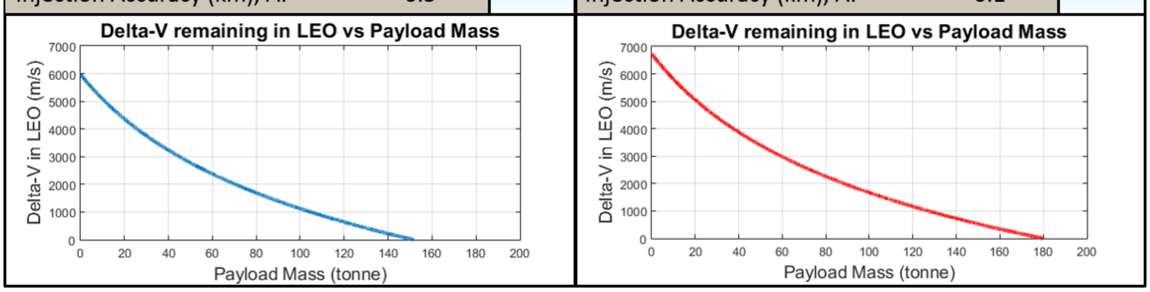


Figure 7. Comparison between launch vehicle concept capability models

Concept 1 is the lower cost and more reliable option, utilizing tried-and-true engine technology that trades efficiency for simplicity. It takes a relatively low amount of time to manufacture the vehicle and prepare it for launch, and the well-understood nature of the technology means that the system is very reliable. Additionally, concept 1 is slightly more structurally robust to variations in environmental conditions (such as temperature and wind speed), and as such has a higher chance of being able to launch as scheduled on any given day i.e. a lower chance of inclement weather causing a delayed launch. This structural robustness also allows concept 1 to accommodate a larger payload fairing because it can withstand higher aerodynamic forces.

Concept 2 is a heavier lift variant utilizing newer engine technology that provides additional efficiency and maneuverability but at the cost of additional complexity, lower reliability, and higher manufacturing and operating costs. The more efficient engines allow this launch vehicle to lift heavier payloads than its less efficient counterpart, ultimately providing more delta-V for similar payload masses. Additionally, the extra controllability provided by the advanced engines offers improved injection accuracy when compared to concept 1. However, the complex nature of the design requires more manufacturing and launch preparation time, and due to the novelty of the technology the system is slightly less reliable (i.e. slightly higher chance of catastrophic failure).

The differences between these two design concepts are not easily summarized. For example, it is difficult to heuristically judge that one vehicle is overall less sensitive to random variation in value (or as one might say, “more robust”) than the other because while concept 1 is less susceptible to weather delays or random failures, concept 2 is less sensitive to precise mission trajectories or unexpected needs for more payload mass. Neither concept is obviously superior to the other. Simulation with consideration of each value-generating scenario (mission in this case) is necessary to determine which concept will produce more value.

Before moving on to the simulation results, a brief discussion will be presented (as in the previous section) in order to clarify the rationale behind certain assumptions and modeling choices. The first of these clarifications concerns the use of the ideal rocket equation to calculate the delta-V vs payload mass curve. The ideal rocket equation is derived with certain implicit assumptions such as constant I_{sp} and no drag from air or gravity. These non-ideal factors primarily affect the vehicle's ascent through the atmosphere, and are accounted for in this example problem by an assumption of a given non-ideal delta-V that must be expended to reach low earth orbit (9600 m/s is assumed). Detailed dynamic simulation of the launch path would be needed to make an assessment of launch vehicle capabilities that more precisely accounts for these non-ideal factors. Once in low earth orbit, these non-ideal factors are assumed to be negligible for the purposes of this example. Additionally, recent work on applying exergy analysis to aerospace systems [70, 71] shows promise and may lead to more comprehensive models of launch vehicle capabilities in the future. Regardless of how it is obtained, the relationship between delta-V and payload mass is necessary to simulate the value of these systems.

Another modeling choice deserving of explanation is the use of a simple binomial probability for modeling the chance of weather delay. It is acknowledged that in a real-world construction of such a model, a more sophisticated representation involving (among other variables) structurally allowable lateral wind speeds for the vehicle and wind speed distributions at the launch site would be desired. However, for demonstrative purposes this representation is enough to loosely model the impact of weather delays on overall system value.

Similarly, a binomial probability is used for the rocket reliability in lieu of a sophisticated simulation of a launch involving fault trees, state variables, etc. A higher fidelity representation would certainly fit within this methodology – the binomial probability is used for demonstrative purposes and simplicity.

4.6. STEP 6: SIMULATE VALUE PRODUCTION

With the mission value model portfolio and the vehicle capability models constructed, all that remains is to evaluate the value of each system. Due to the stochastic nature of the problem, a 3000-trial Monte Carlo simulation of the 30-year lifetime of each concept is used to estimate the probability distribution of total value produced over the lifetime. The simulation proceeds as shown in Table 3 and Figure 8. The results of the simulation are presented in Table 4 and shown in Figure 9. The dark grey bars in Figure 9 denote areas of overlap between the two histograms.

Table 3. Structure of value simulation

Step	Description	Costs Incurred	Value Generated
1	Begin simulation.	Development cost.	N/A
2	The next mission is randomly selected as per the demand percentages, then the simulation waits for the specified vehicle production/rollout time.	Vehicle production and mission costs.	N/A
3	It is launch day. Check to determine if the launch is delayed due to weather. If the launch is delayed, go to step 4. If the launch is not delayed, check to determine if the launch is successful. If the launch is not successful, go to step 5. If so, go to step 6.	N/A	N/A
4	The launch is delayed. Go back to step 3 on the next day.	25% of launch cost.	N/A
5	The launch catastrophically fails. This mission is lost and will not be reattempted. Go back to step 2 on the next day.	Launch and mission failure costs.	N/A
6	The launch is successful. Go back to step 2 on the next day.	Launch cost.	Mission value.

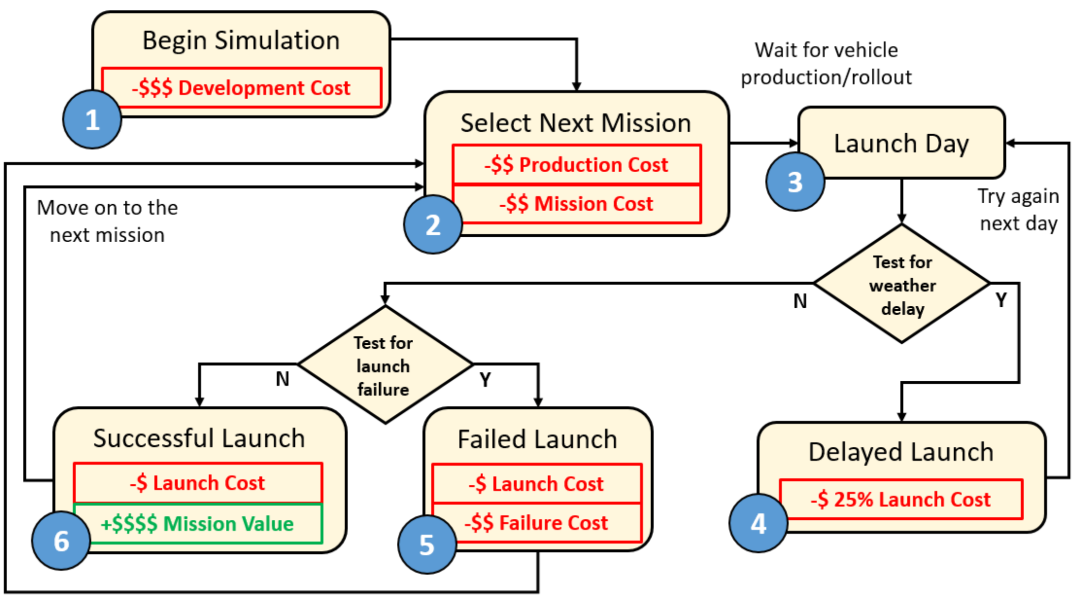


Figure 8. Value simulation flowchart

Table 4. Expected net value and standard deviation of 30-year net value for each vehicle concept

Launch Vehicle	Expected Net Value	Standard Deviation of Net Value
Concept 1, Reliable Mainstay Rocket	$219.0 billion	$24.3 billion, 11.1% of mean
Concept 2, Heavy Experimental Rocket	$194.8 billion	$25.4 billion, 13.1% of mean

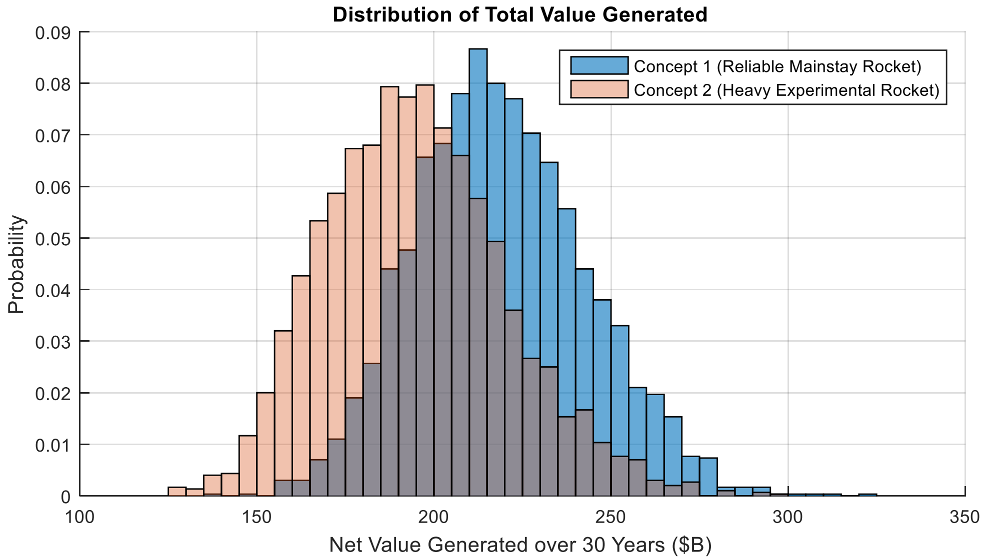


Figure 9. Histogram of the distribution of 30-year net value for both launch vehicle concepts. The dark grey bars denote overlap between the two histograms.

5. USING THE FRAMEWORK TO INVESTIGATE ROBUSTNESS

5.1. INTRODUCTION

The previous section presented a notional example of how this framework might be applied to a systems engineering project, with a publically developed launch vehicle as the system of interest. This section will build upon that example, but with more pointed design scenarios aimed at showing how the framework can be used to answer questions qualitatively related to the concept of robustness, such as "how can the value impact of robustness be determined?" and "how can an inherent preference for risk avoidance factor into the application of this framework?" The intent of this section is to provide guidance to systems engineers and designers who recognize that there are benefits to designing qualitatively more "robust" systems and who want to reap those benefits, but who are unsure of the correct way to go about doing that within a value-driven framework.

5.2. EXAMPLE 1: EVALUATING DIRECT VALUE IMPACT OF ROBUSTNESS

In section 2.4, it was suggested that within a value-driven decision process, the quality of a system often called "robustness" should be taken into account via its impact on value as opposed to via some specific attempt to quantify it directly. The following example will demonstrate how this might look in practice. This example considers a decision scenario in which the baseline design for the launch vehicle has already been established. The lead engineer on the project is faced with a choice between one of two improvements that may be made to the vehicle. Option A is to increase engine efficiency, thereby increasing I_{sp} (and therefore payload capacity). This results in a qualitatively higher performing vehicle. Option B is increase engine durability, thereby significantly improving the vehicle's reliability and resistance to weather. This results in a qualitatively more robust vehicle. The mission value models are the same as those shown in Figure 6, so consequently the capability model takes the same form (AKA the top-level system attributes needed to interface with the value models are identical), with the new alternatives simply having different values for some of the attributes. The decision scenario is shown below in Figure 10, presented as a baseline vehicle capability model with the two possible improvements noted.

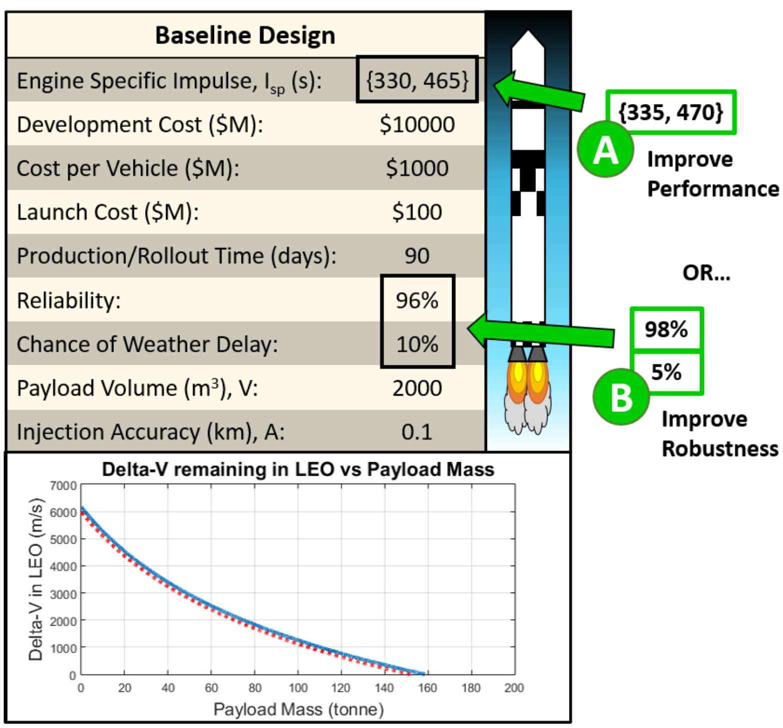


Figure 10. Baseline design for example 1, with possible improvements A and B noted.

As with the example in the previous section, it is difficult to say at first glance which of these concepts should preferred. However, in this case the same difficulty does not arise when considering robustness. At a qualitative level where "robustness" is taken to mean "avoids negative consequences associated with uncertainty," one of these concepts is clearly more robust than the other. However, design decisions are rarely made in a vacuum. Tradeoffs must be considered, and the other design clearly has higher performance.

Robustness is a desired quality in a system, as is performance – some method of direct comparison between the two is needed in order to evaluate which decision alternative is preferred. One approach would be to pick on the basis of intuition, perhaps aided by some visualization technique. Another approach might be to use some quantification scheme to score the systems based on "robustness" and "performance", then use a weighted sum of robustness (as expressed by the score) and performance (as expressed by the score) as the overall score for the system.

However, both of these approaches lack grounding in normative decision theory, leaving little reason for the decision-maker to have confidence that they have made the right decision. In order to determine which option should truly be preferred, a value-driven approach should be used that compares the two in the "same units," so to speak. The capability-based framework can be employed to evaluate this system at a value level. In this way the benefit of each improvement can be quantified, allowing for a decision to be made.

As in the previous example, a 3000-trial Monte Carlo simulation is used to generate a distribution of the vehicle's net value over its 30-year lifetime. The results of the simulation are presented in Table 5 and Figure 11.

Table 5. Expected net value and standard deviation of 30-year net value for options A and B.

Launch Vehicle	Expected Net Value	Standard Deviation of Net Value
Option A, Improve Performance	$206.6 billion	$25.1 billion, 12.2% of mean
Option B, Improve Robustness	$201.0 billion	$23.6 billion, 11.8% of mean

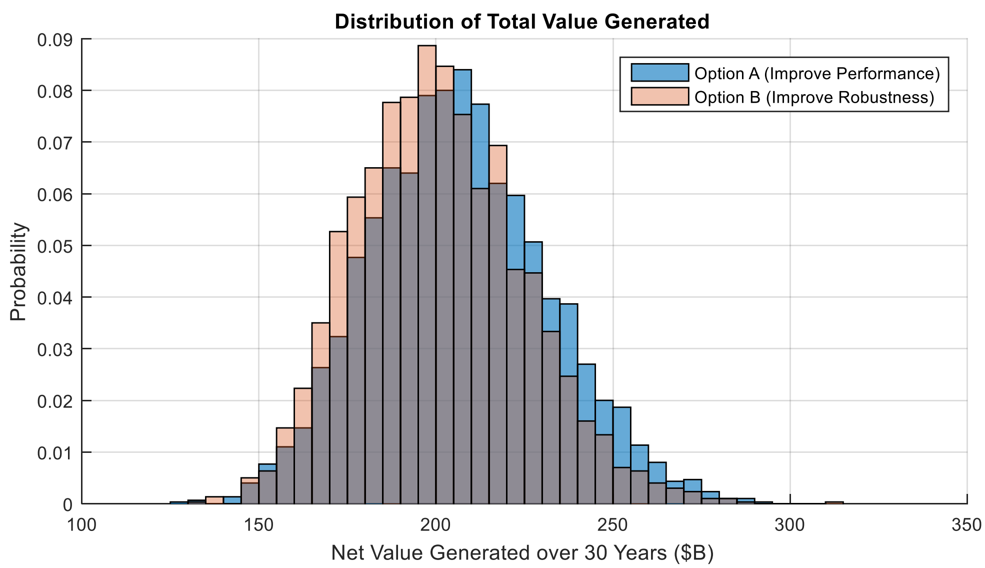


Figure 11. Histogram of the distribution of 30-year net value for options A and B. The dark grey bars denote overlap between the two histograms.

As the results show, in this case it is actually the less robust system that is more valuable and thus preferred. The additional value delivered by the increase in performance outweighs the value added by the increase in robustness.

Note however that the correct conclusion to draw from this example is not "improving performance is always better than improving robustness." This is after all just an isolated example, and one with a fairly unsophisticated simulation and value model at that. Rather, this example simply shows how one can (and should) evaluate the impacts of design changes in complex systems in a value-driven manner when trying to make a decision. Robustness is, of course, a useful quality for a system to have, but as previously mentioned, attempting to quantify robustness directly can have problematic implications from a decision-making perspective. It is better instead to evaluate the value impact of attributes associated with robustness.

To elaborate on this point, an alternate version of this example is presented in which the baseline design concept is slightly less reliable and slightly less resistant to weather – the baseline reliability has changed from 96% to 95% and the baseline chance of weather delay has changed from 10% to 15%. The new decision scenario is shown below in Figure 12.

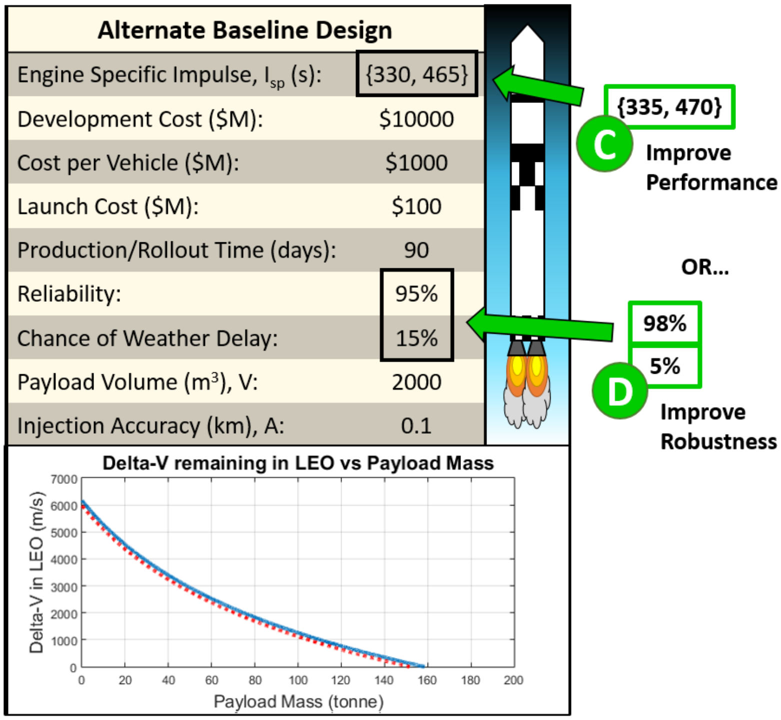


Figure 12. Alternate baseline design for example 1, with possible improvements C and D noted.

At first glance, this is quite similar to the previous decision scenario. Option C is clearly higher performing than the baseline, while option D is clearly more robust. However, the increase in robustness from option D is larger relative to the baseline than in the previous scenario. This causes the value results to change. Table 6 and Figure 13 show the results for this alternate scenario.

Table 6. Expected net value and standard deviation of 30-year net value for alternate options C and D.

Launch Vehicle	Expected Net Value	Standard Deviation of Net Value
Option C, Improve Performance	$200.2 billion	$24.5 billion, 12.2% of mean
Option D, Improve Robustness	$201.2 billion	$23.7 billion, 11.8% of mean

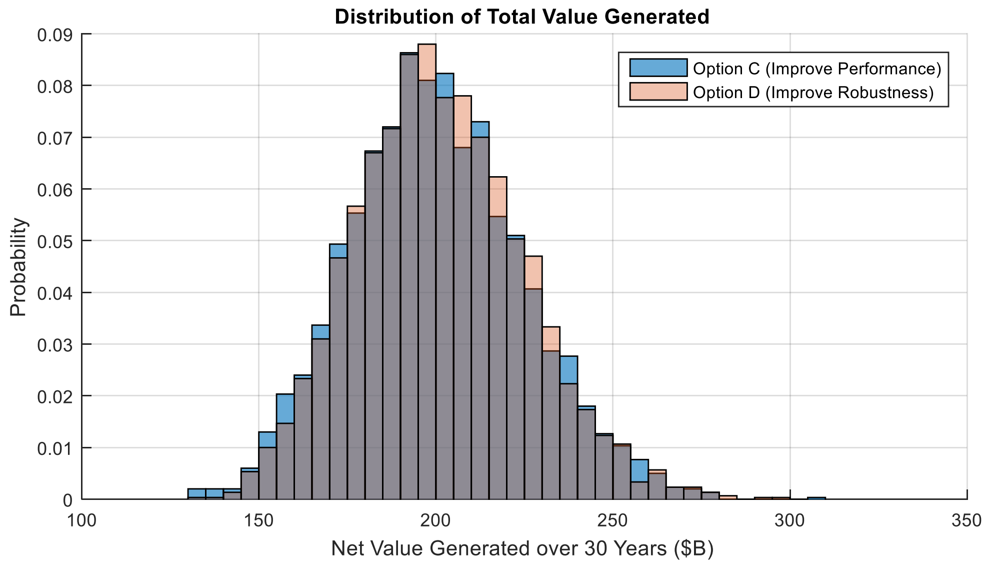


Figure 13. Histogram of the distribution of 30-year net value for alternate options C and D. The dark grey bars denote overlap between the two histograms.

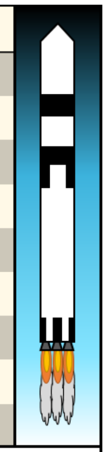
In this situation, the robustness improvement is the more preferred option. Note that this conclusion is reached not because of any inherent preference for robustness over performance or attempt to quantify robustness and then trade it off against performance. Rather, the two options were simply compared on a value basis and a decision made. In general, systems engineers should attempt to quantitatively evaluate the value impact of robustness instead of making decisions based on rules of thumb or intuition.

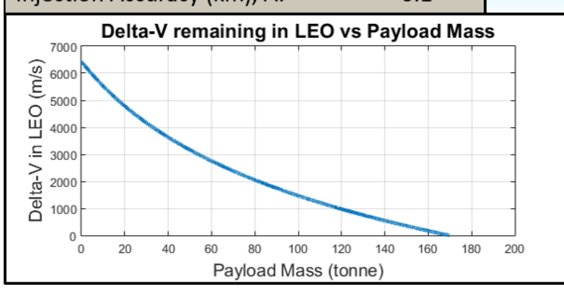
5.3. EXAMPLE 2: APPLYING RISK ATTITUDE TO VALUE

The previous example examined a situation in which the benefits of attributes associated with robustness led to a higher expected value for the system. Although the improvement in value was due to a more reliable system, the decision was still made solely based on which alternative had a higher expected value, not taking any inherent preferences about uncertainty directly into account. If, however, the decision-maker does have some inherent preference for avoiding uncertainty, it is possible to incorporate that into the decision-making process by incorporating concepts from utility theory. An inherent preference for lower variability in value can be represented as a risk-averse risk attitude, and can be applied to the value distributions after they have been determined in order to get a distribution of utility. The following is a somewhat contrived example, but it will serve to demonstrate the point. Consider the two rocket concepts shown below in Figure 14:

Concept 1, Higher Performance Rocket	
Engine Specific Impulse, I_{sp} (s):	{345, 475}
Development Cost ($M):	$10000
Cost per Vehicle ($M):	$1000
Launch Cost ($M):	$100
Production/Rollout Time (days):	90
Reliability:	90%
Chance of Weather Delay:	24%
Payload Volume (m^3), V:	2000
Injection Accuracy (km), A:	0.1

Concept 2, Reliable Rocket	
Engine Specific Impulse, I_{sp} (s):	{330, 465}
Development Cost ($M):	$10000
Cost per Vehicle ($M):	$1000
Launch Cost ($M):	$100
Production/Rollout Time (days):	90
Reliability:	98%
Chance of Weather Delay:	5%
Payload Volume (m^3), V:	2000
Injection Accuracy (km), A:	0.2





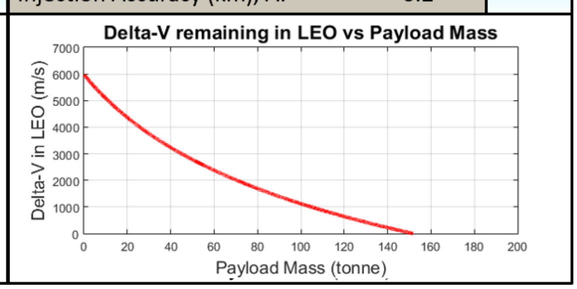


Figure 14. Decision scenario for example 2.

Concept 1 in this case is much more reliable than concept 2, but has considerably lower performance, both in payload capacity and injection accuracy. Concept 2 is considerably higher performing (about 20 more tonnes to LEO), but extremely unreliable with a 10% chance of failure on any given launch and a relatively high chance of weather delay as well. These concepts are simulated (again using the mission value models shown in Figure 6) and the results are shown in Table 7 and Figure 15.

Table 7. Expected net value and standard deviation of 20-year net value for example 2 concepts

Launch Vehicle	Expected Net Value	Standard Deviation of Net Value
Concept 1, Higher Performance Rocket	$128.4 billion	$22.3 billion, 17.4% of mean
Concept 2, Reliable Rocket	$128.5 billion	$19.1 billion, 14.8% of mean

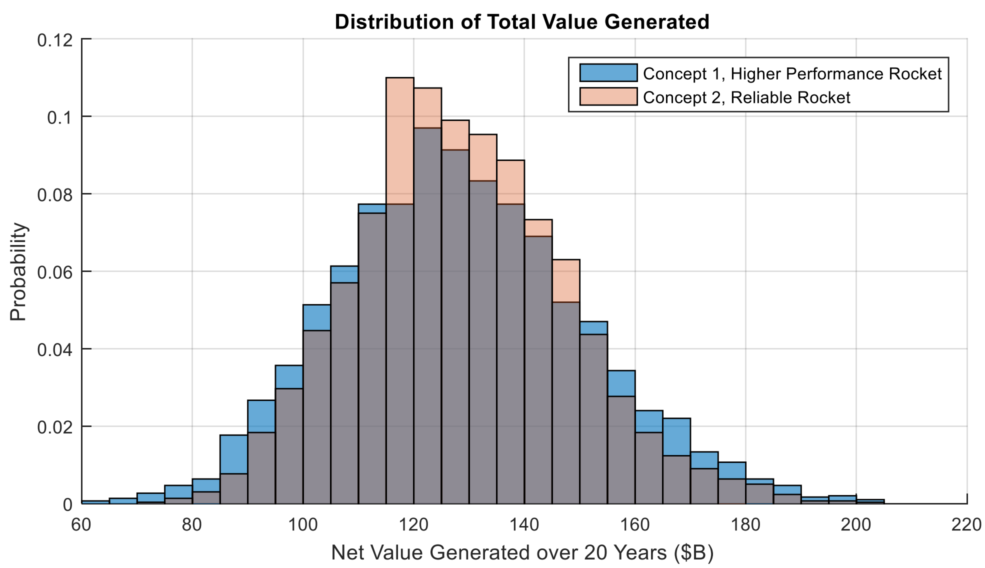


Figure 15. Histogram of the distribution of 20-year net value for example 2 concepts. The dark grey bars denote overlap between the two histograms.

As we can see, the two concepts have nearly an identical expected value. However, the spread of concept 1's value is much higher due to its unreliability.

Suppose the decision-maker in this instance knows intuitively that they prefer concept 2 because it is more reliable, more robust to the chance of catastrophic failure. Utility theory provides a way to formalize this preference mathematically using a risk-averse utility function. There are many risk-averse utility functions, but for this example we will use one of the simplest, the constant absolute risk aversion (or CARA) utility function. The CARA utility function takes the form of an exponential function, and can be elicited using a single certainty equivalent due to the assumption of constant absolute risk aversion. Taking the bounds of the value attribute as \$60-220 billion, and with a certainty equivalent of \$100 billion for the lottery of < \$60B, 0.5, \$220B >, we arrive at the utility function shown below in Equation (3) and Figure 16.

$$u(z) = (1.09575 - 2.73351 \cdot e^{0.01523 \cdot z}) \cdot 10^{9} \tag{3}$$

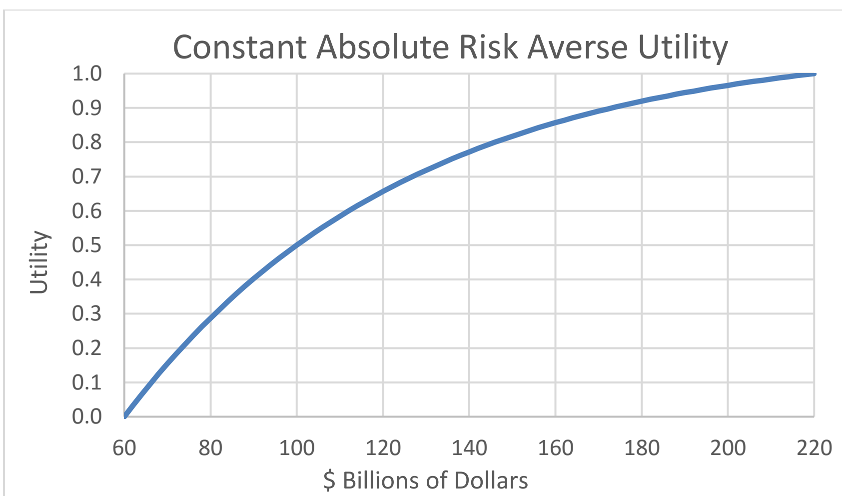


Figure 16. CARA utility function for example 2

Using this utility function, we can transform value into utility, thereby taking the risk preference of the decision-maker into account. Table 8 and Figure 17 show the results.

Table 8. Expected utility for example 2 concepts

Launch Vehicle	Expected Utility
Concept 1, Higher Performance Rocket	0.687
Concept 2, Reliable Rocket	0.694

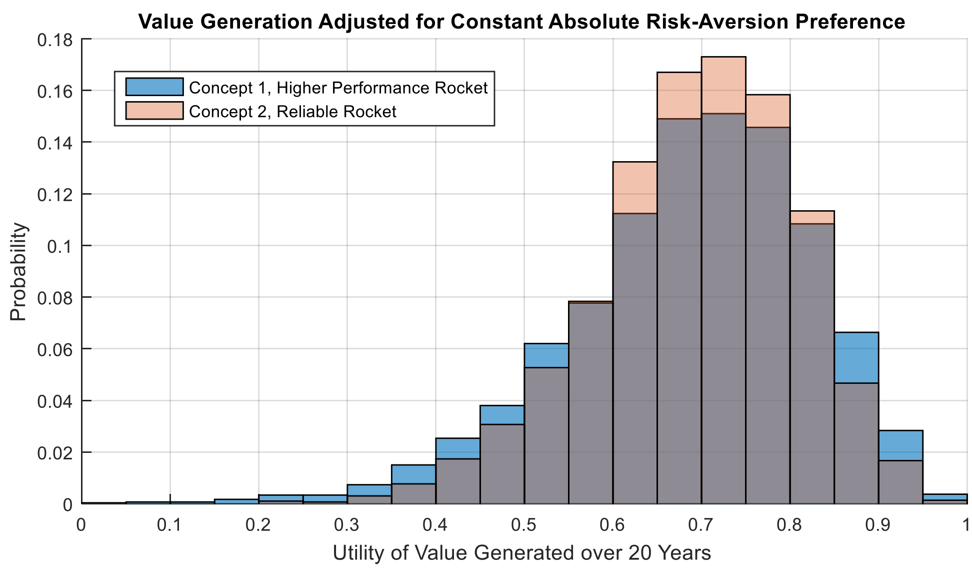


Figure 17. Histogram of the distribution of utility for example 2 concepts.
The dark grey bars denote overlap between the two histograms.

This results in an expected utility of 0.687 for concept 1 and 0.694 for concept 2. Incorporating an inherent desire to avoid uncertainty and risk has caused the preferred concept to become obvious.

It should be noted that some may comment that the realism of this specific example is somewhat questionable. After all, who would knowingly build a rocket with a 10% chance of failure? Some might suggest that such a system should simply be rejected out of hand. This impulse, though natural, should be treated with caution. A basic tenet of normative decision theory is that preferences can and should be represented mathematically such that the preferred decision alternative has a higher expected value or (when risk attitude is involved) a higher expected utility. And indeed, in this example with a risk-averse preference stated, the riskier alternative does end up being less preferred. However, a slight tweak to the concepts could have resulted in the unreliable system actually showing up as more preferred even with the risk-averse utility function in play. Does this mean the value-driven approach to dealing with uncertainty is useless?

In short, no. If a decision-maker in a space agency were to be faced with such a situation and was confident that the result of the value model was out of line with its preferences, this suggests one of two possibilities: either the value model does not fully capture their preferences, or the decision-maker's intuitive preference is actually not in line with what they prefer. If it is the first case – perhaps the decision-maker simply knows that having above a 5% chance to lose a crewed mission throughout the lifetime of the vehicle is absolutely unacceptable – then the penalties for mission failure might need to be higher in order to reflect this. On the other hand, if the numbers assessed as penalties for failure (even on crewed missions) are well-grounded and believed to accurately represent the decision-maker's preferences, it could be that the intuitive assessment is wrong. Systems engineers would be wise to consider both possibilities in such situations.

6. CONCLUSIONS

6.1. FRAMEWORK ADVANTAGES AND DISADVANTAGES

The capability-based value framework offers a number of advantages to systems engineers desiring to use value-driven design to evaluate systems with multiple value-generating scenarios. One of the most prominent of these advantages is that the framework provides a workable alternative to constructing and maintaining a separate model of system capabilities for each value scenario. The approach of this framework consolidates modeling efforts and provides flexibility in the event of mid-design changes in the system's operational context.

Consider the launch vehicle example introduced in Section 4: because the capability model already contains the information needed to interface with new missions, adding an additional mission (or two, or ten) to the mission portfolio could require only the effort needed to construct the new mission value models. In the event of a new mission requiring an aspect or regime of capability not already addressed by the existing capability model, the capability model can be easily modified to address that need. Contrast this with an approach that develops a customized representation of the system for each mission, or that places requirements on certain regimes of system capability that are only relevant to isolated value scenarios, such as specifying a minimum payload to LEO without considering the needs of a Mars mission. Such an approach is less able to accommodate changes in system context. A general capability model that goes beyond a simple vector of attributes to include relationships between attributes is more flexible when these changes occur.

Another advantage of the capability-based framework, as demonstrated in Section 5, is that it provides engineers a means to incorporate a desire for system “robustness” into a value-driven decision making process without having to quantify robustness explicitly. Attempts to quantify robustness, such as Taguchi’s robust design [60] or Kitano’s notion of robustness for biological systems [59], associate the concept with low variability or spread of performance. However, the fundamental reason why most engineers want robustness is to avoid bad outcomes – variance minimization may or may not be a means to achieve this. In light of recent work that has shown through counter examples that variance minimization is not generally consistent with the fundamental desire to be robust to bad outcomes [57, 61], a value-driven approach which directly quantifies the impact of undesired outcomes is more consistent with rational decision making.

More generally, this framework allows for uncertainty in general to be taken into account in a more rigorous and detailed fashion than approaches that use less detailed representations of the system (such as dominance analysis, which has fundamental problems when dealing with uncertainty). The launch vehicle example shown in this thesis demonstrates that simulating the system’s usage throughout its entire life cycle allows for the stochastic evaluation of the system such that the consequences of uncertainties are quantified in its expected value.

This capability-based framework also has some inherent disadvantages. Chief among these is that although it requires comparatively less effort than an approach which maintains multiple competing representations of the system, it does still require considerable effort to implement. For projects which are at the very early stages of implementation, there may not be enough detail available in either the scenario value models or the system architecture to make this value-based approach worth the trouble. In these cases, a more exploratory approach such as that found in tradespace exploration may prove useful. Another disadvantage is that in order to construct this capability-based representation of the system, it must be possible to construct detailed value models for the various value-producing scenarios of the system. Recall the process shown in Figure 4 and described in Section 3 – the value models must come first. Although it is argued in this thesis that this is the correct order in which to construct the models, this could present a problem for large organizations in which there is separation between the engineers dealing with the value scenarios and the engineers working on the system. In these cases, more communication between these teams may be needed if the approach described in this thesis is to be successful.

6.2. FUTURE WORK

There are a number of aspects of this research that could benefit from further investigation and refinement of the ideas. Three notable ways in which the work could be refined are: translating the framework into a SysML schema, refining the launch vehicle example, and demonstrating the approach on an actual systems engineering project. This section will summarize these three major avenues for future work in order to provide guidance to any researchers seeking to improve upon the ideas presented in this thesis or to leverage them for improving systems engineering processes.

One major avenue for refinement is leveraging some of the tools of MBSE, most notably SysML, to present the framework in a language that is more well-known in the systems engineering community. The general approach of this framework is rooted in MBSE principles, so a natural next step for this work would be to take the ideas and process discussed in Section 3 (particularly the structure of the framework shown in Figure 3) and represent them within SysML. A SysML schema for this framework would be an important tool for communicating the ideas expressed in this thesis to the systems engineering community at large. Additionally, translating the framework from the general language used in this thesis into the specific modeling language of SysML would likely lead to new insights regarding refinements, clarifications, and improvements that can be made to the framework itself.

Another avenue for future work is refining the launch vehicle example. This could be done in a number of ways – some examples are:

- Incorporating more detailed mission selection logic
- Adding more stochastic variables into the capability model (or representing variables that were previously expressed with certainty stochastically)
- Increasing the fidelity and diversity of the value models

Each of these will be discussed briefly. Improving the detail of the mission selection logic would involve taking into account factors such as planetary alignment, supply-chain issues, and dynamic agency strategy. This would allow a more realistic picture of the relative utilization of each value scenario than the current "% of total demand" approach allows. Modifying the capability model to represent more variables stochastically would involve using probability distributions to account for various uncertainties in the manufacture of individual launch vehicles. Because the capability model is not limited to an N-dimensional vector of attributes as in tradespace exploration, realistic uncertainties in system capability can be represented and evaluated. Increasing the fidelity and diversity of the value models is the most difficult of these three improvements to implement, primarily because (at the time of this writing) published mission value models that take variables such as payload volume and accuracy explicitly into account when calculating value delivered do not appear to exist. The next step in improving these mission value models should be to scour literature on published mission value models and consult with subject matter experts regarding how these factors might be taken into account in a more realistic mission value model (as opposed to the notional example implementation in this thesis).

These various refinements to the launch vehicle example, though useful, should not be viewed as the end-goal where demonstrating the framework implementation is concerned. That designation belongs to the project of demonstrating this approach by applying the framework to a real systems engineering project – possibly a launch vehicle as well, but using real vehicle data and real mission value models. Improving the fidelity of the launch vehicle example as discussed in the last paragraph is of course a prerequisite for applying this framework to a real launch vehicle. While the most obvious real system for demonstrating this framework (due to the work already performed) is a launch vehicle, any other systems engineering project whose primary artifact creates value through a variety of value-producing scenarios would also be a valuable example. Identifying other systems which may benefit from this approach should be another goal of researchers seeking to carry this work forward. Validating the framework by applying it to an actual systems engineering project (or multiple ones) is a vital step in the process of eventually realizing the benefit of these ideas in systems engineering practice.

6.3. SUMMARY

Value-driven design shows promise as a paradigm for leveraging insights from decision theory to improve systems engineering outcomes. Systems that produce value through a single specific scenario can be evaluated by constructing a value model for that scenario and integrating the model of system capabilities into the value model. However, if a system does produce value through a variety of diverse value-producing scenarios, this should be reflected in the modeling approach. This thesis has proposed that for these systems, a portfolio of value models for the various scenarios should be developed. Then, a general model of system capabilities (or "top-level attributes") should be constructed that can interface with any of the individual scenario value models through a simulation that captures the overall context and all scenarios in which the system produces value. This capability-based framework allows systems engineers to more easily model complex systems in a flexible manner that accounts for the variety in value-producing scenarios.

REFERENCES

[1] P. Collopy and P. Hollingsworth, "Value-driven design," *Journal of Aircraft*, vol. 48, pp. 749-759, 2011.

[2] J. Brathwaite and J. H. Saleh, "On the concept of value and its importance to space systems design and acquisition," *AIAA Paper*, vol. 7866, 2008.

[3] J. Markish and K. Willcox, "Value-Based multidisciplinary techniques for commercial aircraft system design," *AIAA journal,* vol. 41, pp. 2004-2012, 2003.

[4] J. H. Saleh, D. E. Hastings, and D. J. Newman, "Weaving time into system architecture: satellite cost per operational day and optimal design lifetime," *Acta Astronautica,* vol. 54, pp. 413-431, 3// 2004.

[5] M. Ferraro, D. Gorissen, J. P. Scanlan, A. J. Keane, E. Quaranta, B. Schumann*, et al.*, "Toward Value-Driven Design of a small, low-cost UAV," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012.

[6] J. Brathwaite and J. H. Saleh, "Bayesian framework for assessing the value of scientific space systems: value of information approach with application to earth science spacecraft," *Acta Astronautica,* vol. 84, pp. 24-35, 2013.

[7] P. Collopy, "Economic-based distributed optimal design," *AIAA Paper,* vol. 4675, p. 2001, 2001.

[8] S. Castagne, R. Curran, and P. Collopy, "Implementation of value-driven optimisation for the design of aircraft fuselage panels," *International Journal of Production Economics,* vol. 117, pp. 381-388, 2009.

[9] S. Castagne, R. Curran, M. Price, P. Collopy, A. Murphy, and S. Raghunathan, "Value driven design optimization of aircraft structures," in *6TH AIAA Aviation Technology, Integration and Operations (ATIO) Forum, Wichita, Kansas, 25th-27th September 2006, 25th-27th September 2006*, 2006.

[10] O. Brown and P. Eremenko, "Application of value-centric design to space architectures: the case of fractionated spacecraft," *AIAA Paper,* vol. 7869, 2008.

[11] J. Brathwaite and J. H. Saleh, "Value-centric framework and pareto optimality for design and acquisition of communication satellites," *International Journal of Satellite Communications and Networking,* vol. 27, pp. 330-348, 2009.

[12] J. Brathwaite and J. H. Saleh, "Beyond cost and performance, a value-centric framework and pareto optimization for communication satellites," in *AIAA Space*, 2009.

[13] J. Cheung, J. Scanlan, J. Wong, J. Forrester, H. Eres, P. Collopy, *et al.*, "Application of Value-Driven Design to Commercial Aeroengine Systems," *Journal of Aircraft,* vol. 49, pp. 688-702, 2012.

[14] S. Keller and P. Collopy, "Value Modeling for a Space Launch System," *Procedia Computer Science,* vol. 16, pp. 1152-1160, 2013.

[15] M. Mosleh, K. Dalili, and B. Heydari, "Optimal Modularity for Fractionated Spacecraft: The Case of System F6," *Procedia Computer Science,* vol. 28, pp. 164-170, 2014.

[16] P. Collopy and R. Horton, "Value modeling for technology evaluation," *AIAA Paper,* vol. 3622, 2002.

[17] P. Collopy, "Value-driven design and the global positioning system," *AIAA paper,* vol. 7213, 2006.

[18] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*: Princeton university press, 1947.

[19] K. J. Arrow, "Social choice and individual values," *Wiley, New York.,* 1963.

[20] R. L. Keeney and H. Raiffa, *Decisions with multiple objectives: preferences and value trade-offs*: Cambridge university press, 1993.

[21] F. Mistree, D. Muster, J. A. Shupe, and J. K. Allen, "A decision-based perspective for the design of methods for systems design," 1989.

[22] F. Mistree, W. Smith, B. Bras, J. Allen, and D. Muster, "Decision-based design: a contemporary paradigm for ship design," *Transactions, Society of Naval Architects and Marine Engineers,* vol. 98, pp. 565-597, 1990.

[23] D. L. Thurston, "Multiattribute utility analysis in design management," *Engineering Management, IEEE Transactions on,* vol. 37, pp. 296-301, 1990.

[24] D. L. Thurston, "A formal method for subjective design evaluation with multiple attributes," *Research in engineering Design,* vol. 3, pp. 105-122, 1991.

[25] G. A. Hazelrigg, "A framework for decision-based engineering design," *Journal of Mechanical Design,* vol. 120, pp. 653-658, 1998.

[26] G. A. Hazelrigg, *Fundamentals of Decision Making for Engineering Design and Systems Engineering*, 2012.

[27] K. E. Lewis, W. Chen, and L. C. Schmidt, *Decision making in engineering design*: ASME Press New York, 2006.

[28] R. Curran, T. Abu-Kias, M. Repco, Y. Sprengers, P. Van der Zwet, and W. Beelearts, "A value operations methodology for value driven design: medium range passenger airliner validation," in *Proceeding of the AIAA Annual Science Meeting, Orlando 2010*, 2010.

[29] K. B. Marais and J. H. Saleh, "Beyond its cost, the value of maintenance: an analytical framework for capturing its net present value," *Reliability Engineering & System Safety,* vol. 94, pp. 644-657, 2009.

[30] M. A. Khadabadi and K. B. Marais, "Value-Centric Approaches to Design, Operations, and Maintenance of Wind Turbines," in *ASME 2013 International Mechanical Engineering Congress and Exposition*, 2013, pp. V06BT07A081-V06BT07A081.

[31] B. J. Kwasa, C. L. Bloebaum, B. L. Mesmer, H. Kannan, and E. Tibor, "Value Impact of an Organization Structure in the Context of Value-Driven Design," *AIAA Paper,* vol. 1364, 2015.

[32] P. Collopy, "Aerospace system value models: A survey and observations," *AIAA Paper,* vol. 6560, 2009.

[33] K. J. Arrow and R. C. Lind, "Uncertainty and the evaluation of public investment decisions," *The American Economic Review,* vol. 60, pp. 364-378, 1970.

[34] J. H. Saleh, N. C. Jordan, and D. J. Newman, "Shifting the emphasis: From cost models to satellite utility or revenue models: The case for a value-centric mindset in space system design," *Acta Astronautica*, vol. 61, pp. 889-900, 11// 2007.

[35] B. D. Lee, W. R. Binder, and C. J. Paredis, "A Systematic Method for Specifying Effective Value Models," *Procedia Computer Science,* vol. 28, pp. 228-236, 2014.

[36] B. D. Lee and C. J. J. Paredis, "A Conceptual Framework for Value-driven Design and Systems Engineering," *Procedia CIRP,* vol. 21, pp. 10-17, // 2014.

[37] J. A. Estefan, "Survey of model-based systems engineering (MBSE) methodologies," *Incose MBSE Focus Group,* vol. 25, p. 8, 2007.

[38] A. W. Wymore, *Model-based systems engineering* vol. 3: CRC press, 1993.

[39] B. K. Yen, "Pulling it all together: a new model, the enterprise data hub, can present a'single source of truth'without disrupting existing business processes or requiring costly IT reinvestments," *Financial Executive*, vol. 20, pp. 60-63, 2004.

[40] J. Wyatt, "Scorecards, dashboards, and KPIs keys to integrated performance measurement: to effectively measure performance, providers need more than just a balanced scorecard; they also must be able to access performance data from a central data warehouse," *Healthcare financial management*, vol. 58, pp. 76-81, 2004.

[41] M. L. Brodie, "Data integration at scale: From relational data integration to information ecosystems," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 2-3.

[42] O. M. Group, 2015, "OMG Systems Modeling Language (OMG SysML)," http://www.omg.org/spec/SysML/1.4/

[43] O. M. Group, 2015, "OMG Unified Modeling Language (OMG UML)," http://www.omg.org/spec/UML/2.5/

[44] S. Friedenthal, A. Moore, and R. Steiner, *A practical guide to SysML: the systems modeling language*: Morgan Kaufmann, 2014.

[45] T. A. Johnson, J. M. Jobe, C. J. Paredis, and R. Burkhart, "Modeling continuous system dynamics in SysML," in *ASME 2007 International Mechanical Engineering Congress and Exposition*, 2007, pp. 197-205.

[46] D. Broadwell and C. J. Paredis, "Using SysML to Elicit a Value Model in Multi-Stakeholder Value-Driven System Design," in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2014.

[47] A. M. Ross and D. E. Hastings, "The Tradespace Exploration Paradigm," in *INCOSE International Symposium*, 2005, pp. 1706-1718.

[48] A. M. Ross, D. E. Hastings, J. M. Warmkessel, and N. P. Diller, "Multi-attribute tradespace exploration as front end for effective space system design," *Journal of Spacecraft and Rockets*, vol. 41, pp. 20-28, 2004.

[49] V. Pareto, "Manual of political economy," 1971.

[50] G. Agrawal, K. Lewis, K. Chugh, C. Huang, S. Parashar, and C. Bloebaum, "Intuitive visualization of Pareto frontier for multi-objective optimization in n-

dimensional performance space," in *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 2004, pp. 1523-1533.

[51] G. Stump, M. Yukish, J. D. Martin, and T. W. Simpson, "The ARL trade space visualizer: An engineering decision-making tool," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.

[52] G. Stump, S. Lego, M. Yukish, T. W. Simpson, and J. A. Donndelinger, "Visual steering commands for trade space exploration: User-guided sampling with example," *Journal of Computing and Information Science in Engineering*, vol. 9, p. 044501, 2009.

[53] R. J. Malak and C. J. Paredis, "Modeling Design Concepts Under Risk and Uncertainty Using Parameterized Efficient Sets," *SAE International Journal of Materials and Manufacturing*, vol. 1, pp. 339-352, 2008.

[54] C. Bily and R. Malak, "Composing Tradeoff Studies Under Uncertainty Based on Parameterized Efficient Sets and Stochastic Dominance Principles," *SAE International Journal of Materials and Manufacturing*, vol. 5, pp. 418-429, 2012.

[55] R. J. Malak and C. J. Paredis, "Using Parameterized Pareto Sets to Model Design Concepts," *Journal of Mechanical Design*, vol. 132, p. 041007, 2010.

[56] S. W. Miller, T. W. Simpson, M. A. Yukish, G. Stump, B. L. Mesmer, E. B. Tibor, *et al.*, "Toward a Value-Driven Design Approach for Complex Engineered Systems Using Trade Space Exploration Tools," in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2014, pp. V02AT03A052-V02AT03A052.

[57] B. Baxter and R. Malak, "Increasing System Robustness Through a Utility-Based Analysis," in *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2013, pp. V02BT02A020-V02BT02A020.

[58] M. D. Griffin, "How do we fix System Engineering?," in *61st Annual International Congress, Prague, Czech Republic*, 2010.

[59] H. Kitano, "Towards a theory of biological robustness," *Molecular systems biology*, vol. 3, p. 137, 2007.

[60] G. Taguchi, *Introduction to quality engineering: designing quality into products and processes*. White Plains, New York: Kraus International Publications, 1986.

[61] R. Malak, B. Baxter, and C. Hsiao, "A Decision-based Perspective on Assessing System Robustness," *Procedia Computer Science*, vol. 44, pp. 619-629, 2015.

[62] R. C. Mitchell and R. T. Carson, *Using surveys to value public goods: the contingent valuation method*: Resources for the Future, 1989.

[63] B. G. Cameron, E. F. Crawley, G. Loureiro, and E. S. Rebentisch, "Value flow mapping: Using networks to inform stakeholder analysis," *Acta Astronautica,* vol. 62, pp. 324-333, 2008.

[64] K. Arrow and R. Solow, "Report of the NOAA panel on contingent valuation," ed: National Oceanic and Atmospheric Administration Washington, DC, 1993.

[65] P. A. Diamond and J. A. Hausman, "Contingent valuation: Is some number better than no number?," *The Journal of economic perspectives,* pp. 45-64, 1994.

[66] W. M. Hanemann, "Valuing the environment through contingent valuation," *The Journal of Economic Perspectives,* pp. 19-43, 1994.

[67] I. J. Bateman, K. G. Willis, and K. J. Arrow, *Valuing environmental preferences: theory and practice of the contingent valuation method in the US, EU, and developing countries*: Oxford University Press, 2001.

[68] R. T. Carson, N. E. Flores, and N. F. Meade, "Contingent valuation: controversies and evidence," *Environmental and resource economics,* vol. 19, pp. 173-210, 2001.

[69] J. R. Wertz, D. F. Everett, and J. J. Puschell, *Space mission engineering: the new SMAD*: Microcosm Press, 2011.

[70] D. W. Riggins, D. J. Moorhouse, and J. A. Camberos, "Characterization of aerospace vehicle performance and mission analysis using thermodynamic availability," *Journal of Aircraft,* vol. 47, pp. 904-916, 2010.

[71] J. A. Camberos and D. J. Moorhouse, *Exergy Analysis and Design Optimization for Aerospace Vehicles and Systems*: American Institute of Aeronautics and Astronautics, 2011.