

## SEARI Working Paper Series

Title: Flexibility in RSC for a Satellite Radar System

Author: Lauren Viscito

Paper Number: **WP-2009-2-1**

Revision Date: May 15, 2009

Note: This is an excerpt (Chapters 3-4) from Viscito's Masters thesis draft of May 12, titled "Quantifying Flexibility in the Operationally Responsive Space Paradigm"

The content in this paper is in pre-published, draft form. It has not been peer-reviewed and should be used for informational purposes only. The authors warn that the paper may contain typos, misquotes, incomplete, and possibly incorrect content. It is intended that the paper will be revised and elevated to "published" status, at which point the quality caveat will be lifted.

## Chapter 3

# Assessing Flexibility in Tradespace Exploration

*I am a tradespace explorer.<sup>1</sup>*

Tradespace exploration is a philosophical shift from optimal design which aims to give a designer more insight into the trades available and possibly satisfactory solutions off of the traditional utility-cost Pareto Front. Flexible designs tend not to be on this Pareto Front because the inclusion of flexibility enablers, called change mechanisms, adds cost to the initial system, while traditional static analysis does not realize the benefits. Dynamic tradespace exploration is able to realize the benefits of changeable designs, but still lacks a way to identify valuably flexible designs.

Methods for assessing flexibility detailed in Chapter 2 can be confusing, qualitative, not helpful to a designer, domain specific, or not suited to use in tradespace exploration. Having a formal and repeatable metric will assist designers in trading flexibility in systems. Value Weighted Filtered

---

<sup>1</sup>credited to UROP Tim McKinley

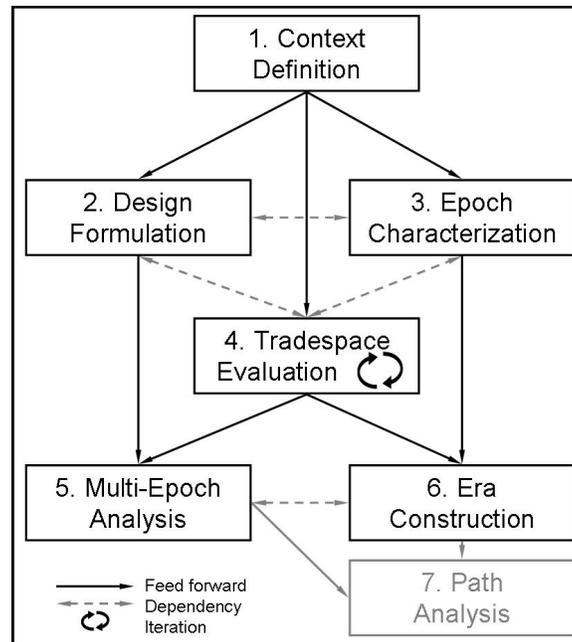
Outdegree is a metric proposed in this research designed to be assessed in a tradespace exploration method. Multi-Attribute Tradespace Exploration (MATE) is a generalizable method that takes user preferences and translates them into designs. MATE was developed to bridge the gap between formal engineering requirements and the creative design process as well as to generalize the design process using computer modeling and utility theory (Ross, 2003).

Generalized Information Network Analysis (GINA) (Shaw, 1998) was developed to compare many design alternatives on a common basis, and used information theory to assess the performance of spacecraft during operations. However GINA had several limitations with regard to modeling Concepts of Operations, in particular the concept of physical translation through launch was difficult to capture with information theory. Launch is often the largest single expense of the space system, and should be included in modeling any new system.

Spaulding (2001) and Derleth (2003) used MATE in two studies: Evolutionary Acquisition as applied to space-based radar and cruise missiles. These studies helped build the extensibility of MATE to designing aerospace systems. Dynamic MATE formalized the process of including time-dependent properties of the system (Ross, 2006).

This chapter describes Dynamic MATE and Epoch-Era Analysis using the seven processes of Responsive Systems Comparison (RSC). RSC is an implementation scheme developed to standardize the application of Dynamic MATE. These processes are as follows:

- 1. Value Driving Context Definition** Identifies system enterprise stakeholders, guides interviews from stakeholders, and aims to determine the important contextual factors that impact the stakeholders.
- 2. Value Driven Design Formulation** The value preferences from the stakeholder are used to produce design trades that are solution neutral.



**FIGURE 3-1:** Block Diagram of RSC Processes.

- 3. Epoch Characterization** The contextual factors are parameterized, and the possible future contexts are identified.
- 4. Design Tradespace Evaluation** The stakeholder requirements, translated into possible designs, are evaluated in a single, static context with utility and cost. All evaluations in this process are during a static context.
- 5. Multi-Epoch Analysis** The performance of many designs is evaluated across many epochs.
- 6. Era Construction** Static contexts are strung together in a storyboard, and the designs are evaluated in dynamic contexts.
- 7. Lifecycle Path Analysis** Strategies for delivering value over the lifetime of the system are developed. (This process is not conducted in the case applications in this thesis.)

This formulation of RSC specifically addresses context changes, addressing uncertainty that may enter into the design performance and evaluation. This set of processes specifically calls out the context changes, making the front end of the process extensive, but leading to better problem

formulation and creating a framework for the computer modeling effort. The steps are described in greater detail in the rest of the chapter, highlighting the steps that are required to compute VWFO.

While each process is described as separate and self-contained, they are highly coupled and in some cases iterative (e.g. Design Tradespace Evaluation). Some processes may also be conducted in parallel (e.g. Multi-Epoch Analysis and Era Construction). Other processes must follow in series (e.g. Design Tradespace Evaluation and Multi-Epoch Analysis). Figure 3-1 shows the dependencies between the process steps. Value Weighted Filtered Outdegree is intended as an additional metric for analysis of designs during the Era Construction step and operationalizes the concepts of flexibility into a usable metric.

For reference, each step is listed in the description by the process number, the type, and step number. For instance, mission statement is referred to as 1.I.2, the second input to the first process.

A simple example is provided to illustrate each step. In the example, Bob needs to determine what size house to build. This simplistic example is not exhaustive or complete, but meant to ground the general RSC framework so that concepts are easier to follow.

### 3.1 Value Driving Context Definition

During this process the purpose of the design problem is identified. The stakeholders begin communicating their needs to the design team, and the value propositions are identified. This process gives context to the subsequent steps. This process focuses on identifying all the interested parties in the design problem.

#### **Process 1**

##### **Inputs:**

- 1.I.1 Value proposition
- 1.I.2 Mission/need statement

**Activities:**

- 1.A.1 Define system enterprise boundary
- 1.A.2 Identify system enterprise stakeholders
- 1.A.3 Identify potential domain experts
- 1.A.4 Identify exogenous stakeholders
- 1.A.5 Elicit exogenous uncertainties
- 1.A.6 Define enterprise stakeholder value network
- 1.A.7 Refine system value proposition(s)

**Outputs:**

- 1.O.1 Value boundary map
- 1.O.2 List of potential domain experts
- 1.O.3 Enterprise stakeholder value network
- 1.O.4 Exogenous system uncertainties
- 1.O.5 Categories of epoch uncertainty
- 1.O.6 System's value proposition(s)

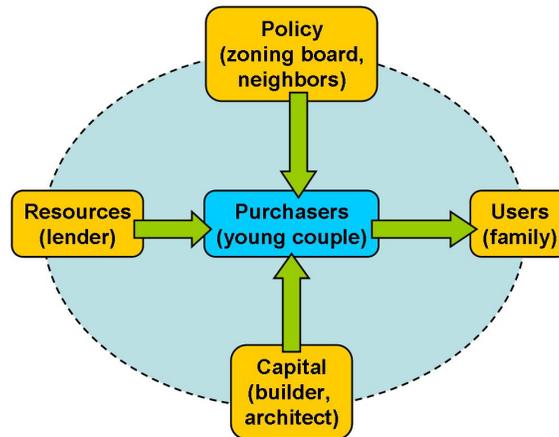
The first and most important process in RSC identifies the purpose of the conceptual design exercise. In this process the stakeholder approaches the designer with a value proposition (1.I.1) and a mission statement (1.I.2). The mission statement should be clear and precise and may be dictated from a higher authority or come directly from the stakeholder. The value proposition is what the user and stakeholders care about. The designer will work closely with the stakeholders to define the system enterprise boundary (1.A.1). Further, within the enterprise there may be more than one stakeholder (1.A.2). It is important to consider multiple stakeholders, because they may have competing value propositions even if they have the same mission statement. The multi-stakeholder problem is difficult because there is no axiomatic way to aggregate their value propositions unless a “dictator” is appointed as a proxy single stakeholder. RSC can stand as a basis for negotiation and compromise, not as a definitive answer to multi-stakeholder aggregation, by allowing various stakeholders to understand each other's value propositions. Even with these limitations, it is important to get as much input from within the enterprise as possible (1.A.6). The enterprise stakeholders may also be able to identify domain experts (1.A.3) who will become helpful when the computer modeling process begins.

Other stakeholders outside the enterprise must also be identified (1.A.4). These external

stakeholders have a direct impact on the design. They may be the end user, investors, or impacted non-users affected by the system. In most space systems the stakeholder responsible for procuring the system is often not the end user but an intermediary agency. During this round of interviews it is useful to begin eliciting the context uncertainties that surround the system (1.A.5).

The value network of the enterprise stakeholders identifies relationships between the various competing stakeholders. After the first round of interviews, the value proposition should be refined if necessary (1.A.7).

### Housing Example



**FIGURE 3-2:** Example Housing Enterprise Value Boundary Map

Bob is building a house. The system boundary includes the house and the land it occupies. The enterprise consists of Bob the stakeholder, the builder, the architect, the town zoning board, the neighbors, and the mortgage lender. This enterprise is illustrated graphically in Figure 3-2, the value boundary map (1.O.1). The builder and architect are also domain experts, able to advise Bob as he forms the system model (1.O.2). Potential epoch uncertainties are limited to the total number of people who will live in the house during the time Bob lives there, i.e. the family in the enterprise

value map (1.O.5). Bob’s value proposition is “to purchase a house that is large enough to be satisfactory and has a large yard, but minimizes the cost of the house” (1.O.6).

## 3.2 Value Driven Design Formulation

Using the inputs from Process 1, this process begins to structure the design problem. Information about the user needs and preferences are formalized in attributes and utility interviews. The stakeholder list is refined, and the design team applies engineering knowledge to build an initial design vector.

### Process 2

#### Inputs:

- 2.I.1 Value boundary map
- 2.I.2 Enterprise stakeholder value network
- 2.I.3 List of potential domain experts
- 2.I.4 Operational concepts / requirements

#### Activities

- 2.A.1 Elicit stakeholder value- and design-space preferences
  - Value proposition(s)
  - Attributes and utility ranges
  - Concept ideas / trade-offs
  - Operational and technical constraints, etc.
- 2.A.2 Define system Concept-of-Operations (CONOPS)
- 2.A.3 Generate system concept(s)
- 2.A.4 Conduct Design-Value Mapping (DVM)
  1. Refine system attribute set
  2. Refine utility ranges
  3. Identify operational system trade-offs
  4. Identify system acquisition trade-offs
  5. Define Design Vector (DV)
  6. Identify potential system “change mechanisms”
  7. Define system “transition rules”
- 2.A.5 Develop initial enumeration range of design vector
- 2.A.6 Record fixed and assumed system parameters

#### Outputs:

- 2.O.1 Elicited Stakeholder(s) attribute and utility Ranges

- 2.O.2 System Concept of Operations (CONOPS)
- 2.O.3 Design-Value Map (DVM)
- 2.O.4 Complied system attributes and utilities
- 2.O.5 Design Vector (DV)
  - Design variables
  - Operations variables
  - Change mechanisms
  - Process variables
- 2.O.6 Initial DV enumeration values
- 2.O.7 “Transition rule” set
- 2.O.8 DV constants
- 2.O.9 Fixed value parameters

During this processes the designer will link the value proposition, which is elicited in solution-neutral user needs, to how the stakeholder will judge alternatives.

The designer will elicit attributes from each stakeholder. Attributes are decision criteria or metrics that are how the stakeholder will determine which designs are best. An attribute set should strive to be complete and perceived as independent (by the stakeholder). Completeness implies that the stakeholder has revealed all the ways he will judge the system. Perceived independence requires that stakeholder will not change preferences on Attribute A because Attribute B changed. It is acceptable for an attribute set to be dependent in reality (e.g. image resolution is inversely proportional to field of regard).

With the enterprise value propositions (2.I.2) from the first process, the designer now begins the formal utility elicitation process (2.A.1). RSC typically uses utility theory to determine the goodness of designs, so the stakeholder must provide utility curves for each attribute they care about. In the case where there are many stakeholders with different value propositions, it may be that some stakeholders do not have preferences for all attributes. At this time the designer should also begin eliciting any constraints and requirements for the system. These will help to shape the attribute ranges.

This research relies heavily on Multi-Attribute Utility Theory, as described by Keeney and Raiffa (1993). In this formulation of classic utility analysis, the stakeholder provides more than one single attribute utility curve (1.A.1). All of the stakeholder single attribute utilities are aggregated in the multiplicative multi-attribute utility form in Equation 3-1. This results in a single metric for each design.

$$KU(X) + 1 = \prod_{i=1}^N [Kk_i U_i(X_i) + 1] \text{ for } K \neq 0 \quad (3-1)$$

$$\text{or } U(X) = \sum_{i=1}^N U(U_i) \text{ for } K = 0 \quad (3-2)$$

where  $K$  is the solution to

$$K + 1 = \prod_{i=1}^N [Kk_i + 1] \quad (3-3)$$

$$\sum_i^N k_i < 1 \quad K > 0$$

$$\sum_i^N k_i > 1 \quad -1 < K < 0$$

$$\sum_i^N k_i = 1 \quad K = 0$$

where

$U_i$  is the single attribute utility

$U$  is the multi-attribute utility

$X_i$  is a single attribute level

$X$  is the attribute vector

$k_i$  is a single attribute weight

$K$  is the overall weighting factor

$N$  is the number of attributes

Utility is not a cardinal scale, which makes it difficult to interpret how much better one design is than another. For instance, meters or feet are on cardinal scales. The difference between two meters and three meters is the same as between three meters and four meters. However, the

difference to the stakeholder between 0.5 and 0.75 utility is not necessarily the same as between 0.75 and 1. Utility orders the designs according to the user preferences. The designer takes the attribute sets and utility sets, and begins to develop a library of system concepts (2.A.2).

The designers come up with a possible set of design variables: those aspects of a design that the engineer can directly affect and “drive” changes in the attribute levels. For instance, physical component choices are aspects the designer can affect, but the resulting performance is not a choice of the designer. The designer then takes the possible set of design vectors and the attributes and conducts design-value mapping, which will help identify those design variables that may drive value. In practice it is often found that there can be too many design variables to feasibly model. In this case only those variables that strongly impact attributes should be varied, and all others should be treated as fixed parameters. Choosing design variables is an iterative process.

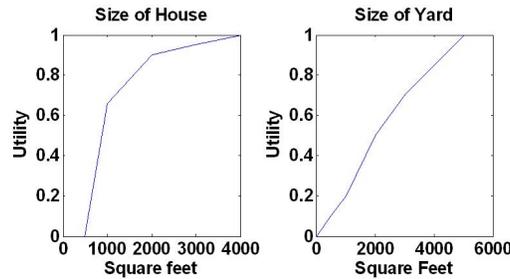
During this process change mechanisms and transition rules (2.A.4) must be identified. Change mechanisms are those design variables that may not drive value, but may make the system more flexible if included (i.e. change mechanisms increase the number of useful transitions from design  $i$  to design  $j$  below a threshold cost). At the same time an initial list of the transition rules must be identified. Transition rules are the specific ways that a system is allowed to change. For instance, a cell phone has many different color covers, and the cover may be changed during the design process, but changing the transmission frequency is not allowed due to constraints placed by licensing.

### **Housing Example**

Using his value proposition, Bob determines two attributes: the size of the house in square feet, and size of the yard (2.O.1). (Although in reality there are many variables that go into choosing a house, some of which may not be easily quantifiable, this simple example only seeks to illustrate

**TABLE 3-1:** Example Housing Attributes and Ranges

Attribute	Range	Units	Weight
Size of House	500 - 4000	$ft^2$	0.6
Size of Yard	0 - 5,000	$ft^2$	0.4

**FIGURE 3-3:** Example Utility Curves

the framework, and so will limit the attributes to two.) The range of the utility curve assigned to each attribute is shown in Table 3-1 (2.O.4). In addition the cost of the house will be estimated.

From the attribute ranges a utility curve is assessed for each attribute (Figure 3-3, and assigned a weighting factor. For simplicity, a linear weighted sum will be used for aggregation.

$$U(x) = \sum_{i=1}^N k_i * U_i(X_i)$$

Bob uses the attributes and performs a Design Value Mapping to several possible design variables (2.O.3). The result of the DVM is shown in Figure 3-4. The DVM includes the enumeration levels of the design variables (2.O.6).

There are several ways that Bob can include change mechanisms in the house, for example building an additional bedroom onto the ground floor or adding a second floor to a single story house. Bob decides to define two transition rules. The first transition rule is adding a floor, subject to the original design being a single story and less than three bedrooms or bathrooms. The second

Design Variable	Level	Attribute	
		Size of House	Size of Yard
# Bedrooms	2,3	9	3
# Bathrooms	1,2,3	3	1
# Floors	1,2	3	9

FIGURE 3-4: Design Value Map

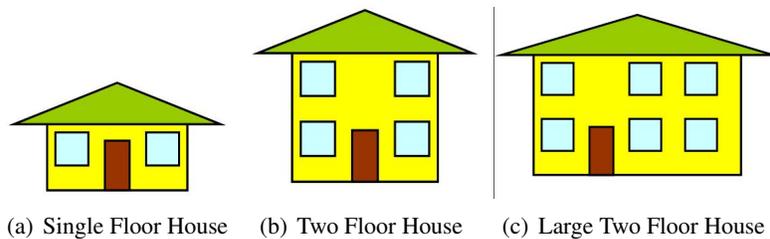


FIGURE 3-5: Example Houses based on Design Vector

transition rule is adding an in-law apartment on the ground floor, subject to the house having only two bedrooms (2.O.7). Since the future size of the family is uncertain, considering possible change mechanisms now may give him flexibility in the future.

### 3.3 Epoch Characterization

Early in the RSC process future uncertainty is captured and parameterized, along with possible design mitigation. By thinking about dynamic states of the system, the design team will understand sooner where the computer model needs to incorporate dynamic contexts. The possible future uncertainties are formalized into an epoch vector.

#### Process 3

##### Inputs:

- 3.I.1 Categories of epoch uncertainties
- 3.I.2 Exogenous system uncertainties
- 3.I.3 Enterprise stakeholder value network
- 3.I.4 Complied system attributes and utilities
- 3.I.5 List of potential domain experts

##### Activities:

- 3.A.1 Interview stakeholders to get future context uncertainties (technical and non-technical)
- 3.A.2 Identify possible future contexts
- 3.A.3 Identify potential uncertainties surrounding stakeholder(s) attributes and utility ranges
- 3.A.4 Define Epoch Vector (EV) and associated constants
  - 1. Map EV to DV
  - 2. Map EV to attributes
- 3.A.5 Record fixed and assumed epoch parameters
- 3.A.6 Define initial enumeration levels for epoch vector

##### Outputs:

- 3.O.1 Epoch Vector
  - Resources
  - Infrastructure
  - Policy
  - Product
  - Technology
- 3.O.2 Epoch Vector constants

### 3.O.3 Fixed epoch parameters

### 3.O.4 Initial EV enumeration levels

Epochs are static snapshots of parameterized future contexts (3.I.1, 3.I.2), stakeholder needs (3.I.3), or mission requirements (3.I.4). Identifying epochs of interest will help the designer to gauge how well the design will perform under many different futures. The inputs to this process include categories of epoch uncertainties (3.I.1), which can include possible technology changes, national policy changes, changing user preferences, and other factors that impact the delivery of system value. The enterprise stakeholder network view is helpful in making sure all relevant types of uncertainty are included. During this process the domain experts (3.I.5) are consulted to inventory risks and opportunities present in that domain.

Defining the Epoch Vector (EV) (3.A.4) is important to the modeling process as it will require modeling all the epoch changes. The EV will appear as constant to all the designs in a single tradespace, but will vary across the tradespaces. The EV should also be mapped to the DV, in case any of the epoch variables directly affect the design variable enumeration levels. For instance, a DV may have three enumeration levels in one epoch, but four in a future epoch due to technology maturing. As a concrete example, there may be three possibilities for processor chips, and as the technology matures, a new processor chip could be considered in the design space. Mapping the epoch vector to the attributes (3.A.4a, 3.A.4b) will eliminate epoch variables that do not affect the attributes, and thus will not affect user value. These can be held constant in the first iteration of the model.

The list of epoch variables is sampled much like the design vector. The levels of epoch variables are listed, and the ones that are assumed fixed are added to the constants list.

The outputs of this process include everything the designer and computer modeler will require to complete the epoch level of modeling.

### Housing Example

Bob realizes that the uncertainties he identified in Process 2 need to be dealt with. He determines that there are several possible scenarios that could occur in the future. To keep the example simple, two epoch uncertainties will be used. The first uncertainty is whether or not Bob will marry and have a child. The second is if Bob's parents will move in with him (3.O.1). Bob conducts another mapping of epoch variables to the attributes, and the result is shown in Figure 3-6.

Epoch Variable	Level	Attribute	
		Size of House	Size of Yard
Child born	no, yes	3	9
Parents move in	no, yes	9	1

FIGURE 3-6: Epoch Variable Mapping to Attributes

## 3.4 Design Tradespace Evaluation

Using the sets of variables generated in previous processes, the design team moves from engineering intuition to a system computer model. The designer then uses the tradespace to explore system design trades.

### Process 4

#### Inputs:

- 4.I.1 Elicited Stakeholder(s) attribute and utility ranges
- 4.I.2 System CONOPS
- 4.I.3 Design-Value Map (DVM)
- 4.I.4 Complied system attributes and utilities
- 4.I.5 Design Vector (DV)
- 4.I.6 Initial DV enumeration values

- 4.I.7 “Transition rule” set
- 4.I.8 DV constants
- 4.I.9 Fixed design value parameters
- 4.I.10 Epoch Vector (EV)
- 4.I.11 EV constants
- 4.I.12 Fixed epoch parameters
- 4.I.13 Initial EV enumeration levels

**Activities:**

- 4.A.1 Use enumerated DV and EV ranges to assist in design of system model architecture and models
- 4.A.2 Build system performance models
- 4.A.3 Sample epoch space (Iterative for each epoch)
  - 1. Sample design space
  - 2. Run performance models over design space
  - 3. Validate technical correctness of model
- 4.A.4 Validate that model covers value space (DVM validation)
- 4.A.5 Analyze performance model results
  - Tradespace plots
  - Conduct sensitivity analysis on results
  - Identify design and utility drivers / trade-offs
  - Calculate Pareto efficient design sets
  - Identify multi-stakeholder compromise designs

**Outputs:**

- 4.O.1 Epoch data sets
  - Design Vector
  - Epoch Vector
  - Constants
  - Attribute and utility values
  - Cost / schedule
  - Intermediate model variables
- 4.O.2 Tradespace plots
- 4.O.3 Key design drivers / trade-offs
- 4.O.4 Utility drivers / trade-offs
- 4.O.5 Sensitivity analysis results
- 4.O.6 DVM validation results
- 4.O.7 Pareto sets
- 4.O.8 Multi-stakeholder compromise designs

Process 4 may take the most time and is often the most difficult for engineers to accomplish correctly because the model is not necessarily a complete representation of the physical system.

Rather, the model is a mapping of fundamental physical relationships necessary to calculate attributes and thus stakeholder value. The detail required of the model is a subjective judgment of the designers, and should carefully balance fidelity of the model against the time required to create that model.

Inputs to this process are all the variable lists created in previous processes as well as the value propositions represented by attributes and utility curves. All these inputs must be taken into account during the modeling process. The first activity in the process is to use the DV and EV (4.A.1) to construct the models, carefully considering the ranges of the vectors. The model must be able to calculate design performance over the full range of design variables and various epochs. It is important to keep in mind these ranges because often they are outside the applicable range of many cost estimating relationships, and therefore many cost estimating relationships (CERs) will not be applicable in these cases. For RSC an estimate is often enough to distinguish designs from one another as long as relative cost is accurate.

Domain experts are required to complete building the system performance models (4.A.2). This is a highly iterative process, as the validation of the models often reveals missing or incorrect physical relationships.

Epoch-Era Analysis is a structured way to model possible future contexts. This process works best when the epoch variables are identified before the modeling effort starts. For instance, if one future uncertainty is whether or not a particular launch vehicle will become available, the computer model needs to include that vehicle.

The model can be abstracted as three levels, each expanding the length of time considered. The first level is a static snapshot of system performance. The middle level is the epoch wrapper, and handles the changes in context around the system. The outermost level is the era level, and links epochs into a sequence to describe future scenarios.

Once the system models are constructed, the enumerated levels of the DVs are used to sample the design space (4.A.3a). This may use a full-factorial expansion of the space, or other sampling techniques. Often the choice of sampling technique is dictated by the time allotted for the model to run. As the designers work to validate the model, a feedback loop may be initiated within this process or to an earlier process.

Once the parametric model has been validated, the designer should return to the value map and make sure that the chosen design variables are actually driving the attributes that the stakeholders have provided (4.A.4). This can be accomplished by using the DVM created in process 2. If it appears that the chosen DVs are not driving value, then the designers need to carefully consider different design variables. Initial analysis of the results should begin if the model is physically correct, and the chosen design variables are driving the attributes.

The primary tool for analyzing results is the tradespace plot. In general, plots depict each design's multi-attribute utility against the estimated cost of the design. Stakeholders should prefer designs with high utility and low cost. A utility of one and cost of zero is the utopia point. This utopia point is usually infeasible, and the stakeholder must consider budget constraints.

Often, there are clear correlations between some design variables and utility levels. These are considered the utility drivers, because the utility is more sensitive to these design variables. Identifying the design drivers and major tradeoffs will be helpful in guiding further analysis and sensitivity studies.

Major design trades occur when there are competing attributes. Often a design may satisfy one attribute to the detriment of another. This may be a fundamental physical relationship constraint, or it may be an imposed constraint, as in the case of technology limits. Another tool for data analysis is sensitivity analysis. Designs may be sensitive to design variable enumeration levels, the epoch enumeration levels, or constants. If designs are particularly sensitive to a constant, it will be

valuable to consider making that constant a variable. The utility-cost Pareto Set is the set of designs for a static tradespace that are non-dominated. That means, for a particular cost, the Pareto Efficient design has the highest utility. The Pareto Set designs are of particular interest to the designers and stakeholders (4.A.5) as these designs are traditionally considered the ‘best’ in the design space.

The outputs of this process include all the static model outputs, including performance data, and intermediate variables and analysis.

### Housing Example

Bob now needs to model the house. Again, to keep the example simple, two parametric equations are used to model the relation of the design variables to the attributes, and one to estimate the cost of the house. The parameters for the equations are estimated, for instance, the size a bedroom was assumed to be  $150 \text{ ft}^2$ , while a bathroom added  $50 \text{ ft}^2$ , and a second floor doubled the square footage of the equivalent single floor house. To estimate the cost of the house, a fictional monthly payment, also based on the size of the house was assumed, and 12 payments multiplied by 15 years gives an approximation of the cost.

$$\begin{aligned} \text{House Size} &= (500 + 150 * \text{Beds} + 50 * \text{Baths}) * \text{Floors} \\ \text{Yard Size} &= 1200 - \frac{\text{House Size}}{\text{Floors}} \\ \text{Estimated Cost} &= (300 * \text{Beds} + 100 * \text{Baths}) * \text{Floors} * 12 * 15 \end{aligned}$$

This is of course a very rough approximation of the square footage and possible payments on a house. However, the trends are the important part, and that the attributes move in the right direction. For instance, a larger house will cost more (4.O.3). The tradespace is generated by running the DV through the three equations. A design space of twelve designs results from a full

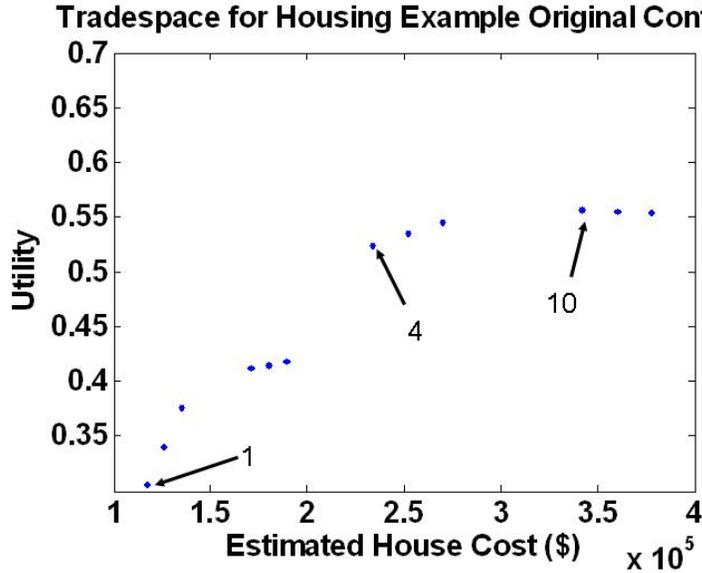


FIGURE 3-7: Example Housing Tradespace

TABLE 3-2: Pareto Efficient Designs for Baseline Context

Design Number	Num of Beds	Num of Floors	Num of Baths	Utility	House Size ( $ft^2$ )	Yard Size ( $ft^2$ )	Cost (\$K)
1	2	1	1	0.3052	850	350	117
4	2	2	1	0.5248	1700	350	234
10	3	2	1	0.556	2000	200	324

factorial enumeration of the three design variables. The tradespace shown in Figure 3-7 has many designs in the Pareto Front, which is in the upper left corner of the tradespace, and some of these designs are shown in Table 3-2.

Bob must trade between a higher utility design such as Design 10 which carries a higher purchase cost, and a slightly lower utility design with a less expensive purchase cost. However, these results assume that the context of this tradespace will hold true forever. If the uncertainties identified earlier manifest, Bob's preferences on the attributes may change. For instance, if Bob has a child, they may now want at least 1000 square feet of space, but would place more weight on having a large yard (4.O.1); this preference change can be seen illustrated in Figure 3-9.

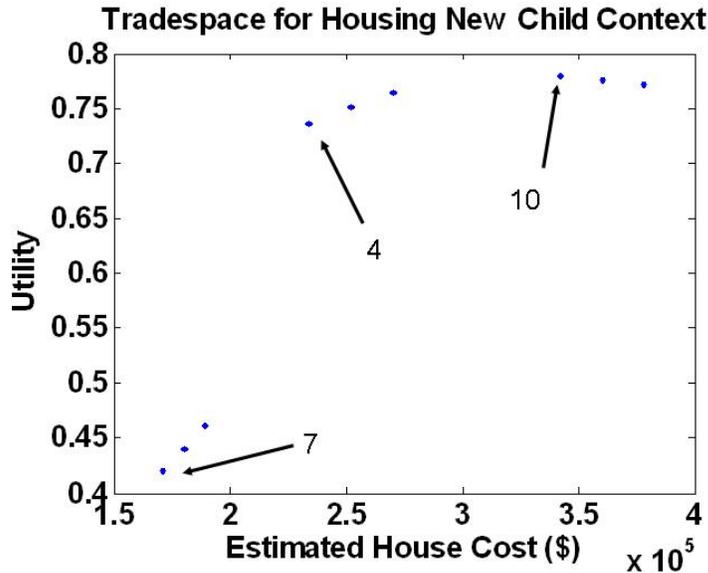


FIGURE 3-8: Example Housing Tradespace Under New Child Context

TABLE 3-3: Pareto Efficient Designs for New Child Context

Design Number	Num of Beds	Num of Floors	Num of Baths	Utility	House Size ( $ft^2$ )	Yard Size ( $ft^2$ )	Cost (\$K)
4	2	2	1	0.3444	1700	350	234
7	3	1	1	0.024	1000	200	171
10	3	2	1	0.384	2000	200	324

Now the range on the attribute square feet is changed, from  $500\ ft^2 - 4000\ ft^2$  to  $1000\ ft^2 - 4000\ ft^2$ . When the tradespace is run again, it appears as in Figure 3-8. Designs have been eliminated from consideration because they no longer meet minimum attribute levels, and designs that were Pareto Efficient before are no longer on the Pareto Front.

In the second context change Bob considers how his needs will change if his parents need to move into the house with him. In this case, he would like to have a some extra space, but no longer needs to have a large yard, so the weight of that attribute is much less. Square feet in the house becomes more important and is weighted at 0.8, while the yard is weighted at 0.2. When the design space is evaluated under these new conditions, the tradespace appears as in Figure 3-10. This epoch has

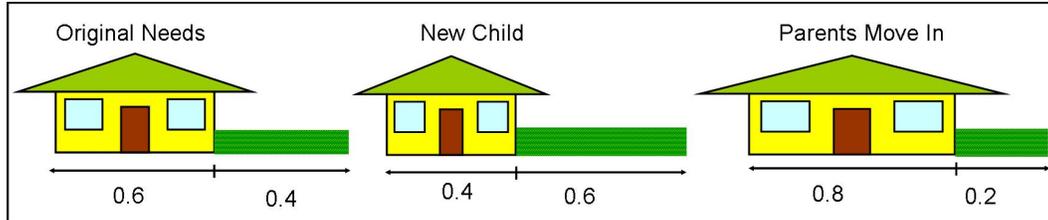


FIGURE 3-9: Illustration of Epoch Preference Changes on Attribute Weights.

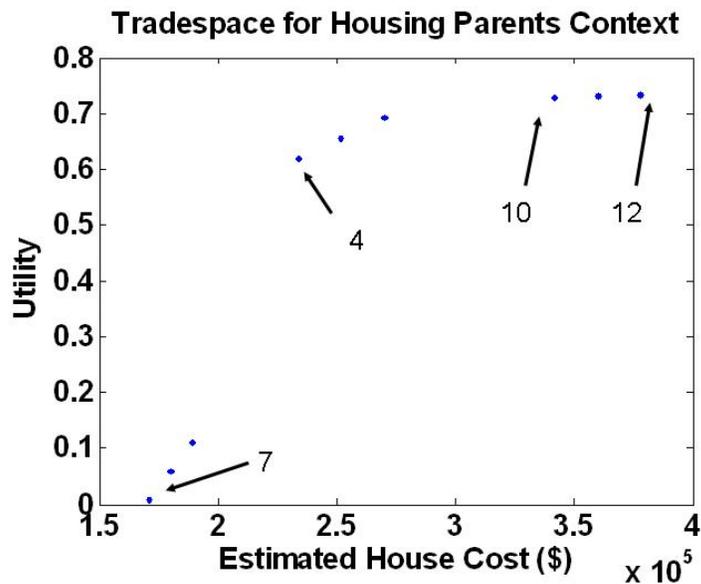


FIGURE 3-10: Example Tradespace for Parent Context

many more designs on the Pareto Front, in particular, the very expensive, large designs move to the Pareto Front.

### 3.5 Multi-Epoch Analysis

Now that the designer has a tradespace for each epoch, and has analyzed the behavior of the design space in a static context, it is time to consider epoch-spanning analyses. The designs are evaluated across many epochs, looking for trends in system behavior.

**TABLE 3-4:** Pareto Efficient Designs for Parent Context

Design Number	Num of Beds	Num of Floors	Num of Baths	Utility	House Size ( $ft^2$ )	Yard Size ( $ft^2$ )	Cost (\$K)
4	2	2	1	0.6188	1700	350	234
7	3	1	1	0.008	1000	200	171
10	3	2	1	0.728	2000	200	324
12	3	2	3	0.732	2200	100	378

**Process 5****Inputs:**

- 5.I.1 Epoch data sets
- 5.I.2 “Transition rule” set

**Activities:**

- 5.A.1 Generate transition matrices for each Epoch data set
- 5.A.2 Calculate tradespace / epoch statistics
  - 1. Pareto Trace
  - 2. Filtered Outdegree (FOD)
  - 3. Tradespace yield
- 5.A.3 Identify highly passive value robust designs
- 5.A.4 Identify highly changeable designs

**Outputs:**

- 5.O.1 FOD function per design, per epoch
- 5.O.2 Pareto Trace numbers
- 5.O.3 List of passively value robust designs
- 5.O.4 List of highly changeable designs
- 5.O.5 Tradespace yield statistics
- 5.O.6 Epoch transition matrices

The inputs to this process include the epoch data sets produced in the previous process, including all the attribute performance and utility data, as well as cost and schedule for each design. At this point the change mechanisms that were included in the design are assessed. The transition rules are applied to each design.

### 3.5.1 Pareto Trace

A Pareto Efficient set of designs, or those designs that are non-dominated, can be determined from the utility-cost plot for each epoch. A Pareto Efficient design has the highest utility of all other designs at a given cost, and represents a best “value” design. The Pareto Trace of a design is the “number of Pareto Sets containing that design” (Ross et al., 2009). Designs that have very high Pareto Trace are said to be passively value robust. These designs have the ability to maintain high utility over changing epochs. After all epochs are evaluated, the Pareto Trace for each design can be normalized by the number of epochs evaluated. This is called a Normalized Pareto Trace (NPT) (5.A.2a).

### 3.5.2 Filtered Outdegree

Previous work has assessed changeability in tradespaces through calculating the number of possible transitions to alternate design configurations a given design can make within the tradespace. When viewed as a tradespace network, this is the ‘outdegree’ of a design. Only counting the transitions available at an acceptable cost results in ‘filtered outdegree’ or FOD (Ross et al., 2008). This captures the changeability of the design. FOD may also be normalized by the highest FOD result so that analyses among different subsets are comparable.

Transition rules tell the designer whether it is permissible or feasible to transition to another design from the current one. If a change is permitted, an arc in a transition network graph is formed, which includes all the designs in the tradespace. At the same time the cost for each transition and the time it will take to complete it are calculated and stored. Filtered Outdegree is the number of transitions that can be made from a given design below a certain cost filter (5.A.1, 5.A.2b).

Tradespace Yield is another figure of merit for the tradespaces (5.A.2c). It indicates how many of

the designs for a given context are feasible. A feasible design is both technically feasible and delivers acceptable levels for each attribute.

At the end of the process, the designer should have a set of metrics for each design: Filtered Outdegree, Pareto Trace, and Tradespace Yield. The designer also has the transition networks required to complete the later processes.

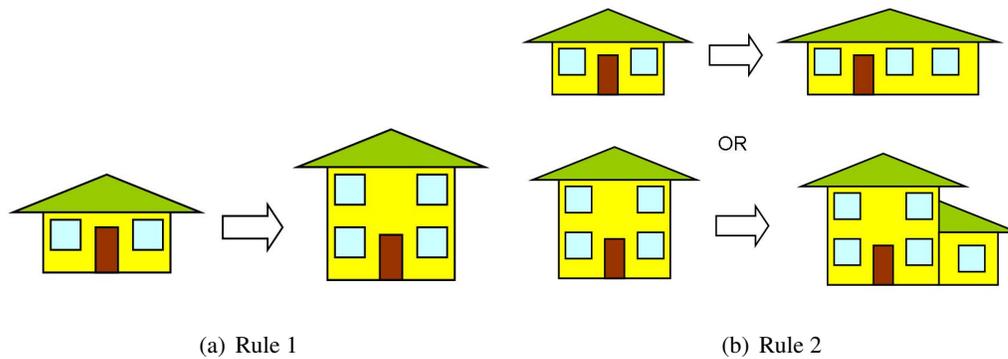
### **Housing Example**

Bob realizes that there is no obvious solution to his decision problem when changing epochs are considered. He decides to examine the dynamic properties of the design alternatives. The Pareto Trace can determine the passive value robust designs. This entails counting the number of times a design appears in the Pareto set of an epoch. As there are three epochs, the maximum possible Pareto Trace is three. The tradespace yield is the number of designs that are feasible for each epoch. The tradespace yield for the static context is 12 designs, for the new child epoch it is 9 designs, and for the parents epoch it is 9 designs (5.O.5).

Bob then applies the transition rules to the design space to see if there are any changeable designs available. The transition rules are applied as follows:

- Rule 1 A second floor may be added if the design has one floor, two bedrooms and one or two baths. Doing so will add a bedroom, a bathroom, and cost 0.5% of the initial purchase price without impacting the size of the yard.
- Rule 2 An addition may be built on the ground floor if the design has two bedrooms. Doing so will add a bedroom, will cost 0.25% of the initial purchase price, and will reduce the size of the yard.

These transition rules are simplistic approximations of the types of changes that can be made to a house, and are illustrated in Figure 3-11. Bob has decided he could not spend more than \$75,000



**FIGURE 3-11:** Illustration of Transition Rules

on any changes to the house. When the rules are applied to the static tradespace, the Filtered Outdegree results are shown in Table 3-5.

From the Pareto Trace results, it looks like design 4, 7 or 8 would be passively value robust (5.O.3). Designs 1 and 2 are highly changeable, with two transitions available below the threshold cost of \$75,000 (5.O.4).

### 3.6 Era Construction

While Process 5 considers the body of epochs as a whole, this process considers time-ordered sets of epochs. The designer can evaluate the design space in a possible future scenario, applying long-term objectives to the system.

#### Process 6

##### Inputs:

- 6.I.1 Sampled Epoch Vector (EV) space
- 6.I.2 Future uncertainty characterization
- 6.I.3 Long-term strategies
- 6.I.4 Forecasts
- 6.I.5 Epoch data sets

**TABLE 3-5:** Pareto Trace and Filtered Outdegree for Housing Example

Design Number	Pareto Trace	Filtered Outdegree
1	1	2
2	1	2
3	1	1
4	3	1
5	3	1
6	3	1
7	3	0
8	3	0
9	3	0
10	3	0
11	1	0
12	1	0

## 6.I.6 Transition networks

**Activities:**

- 6.A.1 Choose scenario development approach(es)
  1. Define given era start and end points, maximum number of epochs in given era
  2. Scenario development / planning considerations for era construction
  3. Potential epochs available at a given point in time
  4. Probability epoch is activated at a given point in time
- 6.A.2 Path dependency between epochs
- 6.A.3 Epoch durations
- 6.A.4 Sample potential era space for range of possible futures by constructing Eras given considerations and approach
- 6.A.5 Determine Value Weighted Filtered Outdegree

**Outputs:**

- 6.O.1 Set of eras: each an ordered set of epochs with durations
- 6.O.2 VWFO for all designs
- 6.O.3 Era spider plots

Era construction is how RSC characterizes possible futures. The primary method to generate eras in this process is scenario formulation (6.A.1). Domain experts are useful for creating eras with a high probability of occurring. The inputs to the process are the sampled epoch space, which may be sampled in the same way that the design space was sampled or another method as appropriate. The designer needs to assign the era start and end points. There is also a probabilistic approach,

where the following epochs are dependent on previous epochs. Scenario creation is often the most familiar to decision makers and may be easiest to explain and understand (Roberts et al., 2009).

### 3.6.1 Assessing Flexibility - Value Weighted Filtered Outdegree

Once the era has been identified, the designer can begin to look for valuably flexible designs.

Because the value of flexibility is only realized in the presence of uncertainty, the designer needs to have a possible future era in which to assess the design in the tradespace. The major contribution of this research to Dynamic MATE is described here: Value Weighted Filtered Outdegree, defined as:

$$VWFO_i^k = \frac{1}{N-1} \sum_{j=1}^{N-1} [\text{sign}(u_j^{k+1} - u_i^{k+1}) * \text{Arc}_{i,j}^k] \quad (3-4)$$

where

$N$  is the number of designs considered

$k$  is the current epoch

$k + 1$  is the next epoch in the era

$i$  is the design under consideration

$j$  is the design to be transitioned to

$u_i^{k+1}$  is the utility of design  $i$  in the  $k + 1$  epoch

$u_j^{k+1}$  is the utility of the design  $j$  in the  $k + 1$  epoch

$\text{Arc}_{i,j}^k$  is the logical value indicating the presence of a transition arc from design  $i$  to design  $j$

From the information gained by looking at highly changeable (FOD) or passively value robust (NPT) designs, a metric was developed that incorporated the possible net utility change of a changeable design, and guidelines for where to look in the design space to find these designs. By being computationally simple and reusing metrics that are calculated in the course of RSC, this metric takes advantage of the work already done during tradespace exploration.

$VWFO_i^k$  is dependent on the choice of  $N$ , the subset of designs from the design space. The analyst can choose to look at the VWFO of an entire design space, in which case  $N$  is the same as the total

number of designs in a the tradespace study. Alternatively, a smaller subset of designs can be chosen, and examined in great detail. VWFO uses the direction change in utility to determine if a particular transition is ‘good’, which occurs when the design transitions to a design of higher utility. By summing both the positive and negative transitions, the designer can see designs that are valuably flexible, (the design with positive VWFO), and the designs that are changeable but are carrying ‘dead weight’ (the design with negative VWFO).

It may appear that a better indication of valuable flexibility could be gained if only positive utility transitions are included in the sum of  $VWFO_i^k$ . In the current form VWFO sums all transitions, both positive and negative. This means that designs that have many transitions (the highly changeable designs that may be identified through FOD) with approximately equal numbers of ‘good’ and ‘bad’ transitions will have a VWFO of close to zero. This appears to unfairly punish the designs for having transitions to lower utility designs. If only positive utility transitions were allowed designs with many transitions would perform very well. However, there are some problems with that form of VWFO.

Unlike FOD, which considers changeability in a static context, VWFO overlays a change in context on the design space. The metric seeks to identify designs that will have many positive transitions in that specific context change. Therefore, being able to identify designs that have many positive utility transitions and few negative utility transitions is advantageous to the designers. The additional logic overlay that comes from the context changing allows the designer to designate designs that have ‘extra’ changeability. This ‘extra’ changeability is not valuable *in this context change* and therefore those designs are not valuably flexible in this particular context change. (But during a different context change those designs might be valuably flexible.) Therefore, one may think of ‘positive utility transitions’ available as offsetting ‘negative utility transitions’ as a way of using short term gains to offset possible long term gains (i.e. the negative utility transitions may become positive utility transitions in a different context change).

**TABLE 3-6:** Housing Example Value Weighted Filtered Outdegree

Design Number	Epoch Transition	
	Static to Child	Child to Parents
4	0.0396	0.1092
5	0.0224	0.0748
6	0.0052	0.0404

In addition, only counting positive utility transitions eliminates a portion of the design space from analysis. If a constrained epoch exists, such that the valid design space is very small, the designer may want to know which designs are changeable without being valuably flexible. During the last process of RSC, Lifecycle Path Planning, the possible design trajectories may be limited if only positive utility transitions are considered, thereby limiting the usefulness of the process. Further research is required to determine if this concern is valid. Appendix A provides more discussion on the issue of allowed utility transitions.

### **Housing Example**

The Era chosen for analysis is an ordered set of the static context, then the ‘new child’ epoch, and finally the ‘parents move in’ epoch (6.O.1). Bob at first needs a small house, as he can only afford relatively small monthly payments. When his child arrives he needs more space, but when the parents move in he can afford larger monthly payments. One way to achieve all his goals in this era is to look for a flexible house design that will allow him to make valuable changes. VWFO was applied to the era and the results are shown in Table 3-6 (6.O.2).

Based on the results of this flexibility analysis, Bob would be wise to purchase house design 4, which is a smaller house with the most valuable flexibility if he needs to expand to accommodate a growing family. While designs 1 and 2 had more changeability, those designs were not valid during the second two epochs, which would result in a period of time where Bob was not satisfied

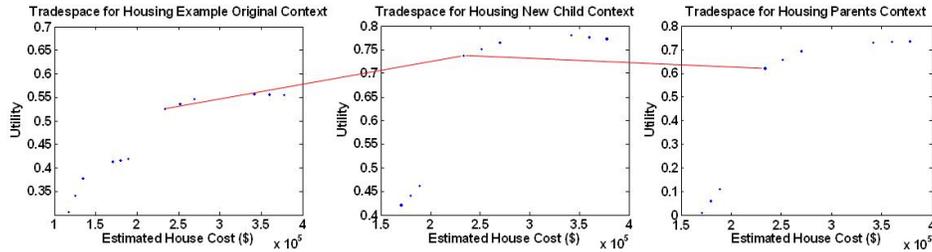


FIGURE 3-12: Housing Example Era with Design 4 Trace

with his house. This simplistic example shows how the RSC framework can be applied to a design decision problem.

### 3.7 Lifecycle Path Analysis

This process uses the eras constructed in the previous process and applies different dynamic strategies to determine design trajectories. This process is not conducted in this thesis, but is described here for completeness of the RSC framework.

#### Process 7

##### Inputs:

- 7.I.1 Era Sets
- 7.I.2 Epoch data sets
- 7.I.3 Epoch Transition Matrices
- 7.I.4 Compiled System Attributes and Utilities
  - Time dependent preferences
  - Programmatic utility
- 7.I.5 Across-Era system trajectory objective

##### Activities:

- 7.A.1 Formalize Program-level utility expectations
- 7.A.2 Calculate utility trajectories of designs across Eras
- 7.A.3 Find “Best Paths” across Eras using appropriate strategy
  - Minimize distance from utopia trajectory
  - Maximize system value at least cost
  - Maintain cost
  - Maintain system value

- Minimize value outages
- 7.A.4 Calculate distance from utopia trajectory (Active value robustness metric)
- 7.A.5 Identify evolution patterns across multiple trajectories for real options investment suggestions in program

**Outputs:**

- 7.O.1 System evolution strategies mapped to “best path” trajectory
  - List of designs
  - Execution transitions
  - Execution times
  - Execution costs

The purpose of the Lifecycle Path Analysis process is to develop near- and long-term system value delivery strategies in response to time-dependent contextual uncertainties (described via Era timelines).

Path analysis takes as inputs the eras of interest (7.I.1) and the epoch data sets for those eras (7.I.2). The designer must have a cross-era objective (7.I.5) in mind to perform path analysis. This could be using passively value robust designs to retain utility across the era, or it could be a more complicated changeability strategy, in the case when no designs are passively value robust over the entire era.

The designer can take this era objective and calculate the utility trajectory (7.A.2) of a design over the era. The designer can then use the objective to determine the ‘best path’ across the era (7.A.3) whether this means following the design that achieves utility closest to one, finding the least expensive design that maximizes utility over the era, or any of several other strategies. If the design is changeable and transitions are executed the designer can determine the distance from the utopia trajectory (a theoretical trajectory with utility of one in every epoch) and find the active value robustness (7.A.4).

Out of this process the designer will gain an understanding of potential strategies for maintaining value over the lifetime of the system. The possible ‘best path’ trajectories will result in a list of

designs in the trajectory, any transitions that have to be executed to maintain the trajectory, the transition execution costs and the time to achieve the changes (7.O.1).

### **3.8 Limitations of Responsive Systems Comparison**

There are several limitations to RSC. It relies heavily on stakeholder inputs, that many stakeholders may not be used to and may not desire. The long and complicated nature of the process is designed to remove solutions during the problem formulation. For a designer who is not familiar with RSC, it is difficult to wait until everything is in place to begin modeling. It is easy to impose solutions on the concepts too early in the conceptual design. By imposing solutions onto the problem space, the solution space becomes contracted.

Stakeholder input is absolutely essential to the process, but most customers are used to throwing requirements over a wall to the designers and not thinking any more about it until the design review. The expectations of the stakeholders and customer need to be carefully managed to make sure that involvement and feedback is timely. Often, the stakeholder with decision making power is very busy, and conducting extensive interviews over time is not feasible. In these cases, a proxy stakeholder must be used. Because utility curves are axiomatically only valid for the people who give them (Keeney and Raiffa, 1993) there are errors built into the value proposition anytime the utility team has to resort to proxies. This can be mitigated by doing sensitivity analysis on the results. Another way to combat these errors is to build the model in such a way that assessing utility is a quick post-processing task.

Another limitation is the parametric nature of the models typically used to produce the tradespace. Outcomes are heavily dependent on the modeling effort, and conveying the multitude of assumptions and estimations in a typical model to the stakeholder is very difficult. The models are approximations of a real design. Often the parametric models are sufficiently detailed to

distinguish the design and value drivers. However, many times the stakeholders are willing to put more faith in the model than they should. To combat undue reliance on model outcomes, the assumptions and limitations in the model must be abundantly clear.

### 3.9 Summary

RSC is a conceptual design framework that enables designers to examine many different designs on a common basis. Typically, designs are plotted in a utility-cost space that can reveal trends in the performance attributes and can give designers insights into design trades available (Ross et al., 2004). These tradespaces are a static snapshot, reflecting stakeholder preferences on system performance. All systems exist within a defined context and as the context changes, the stakeholder preferences may change, the environment in which the system operates may alter, or both. A period of time with a fixed set of preferences and environment for the tradespace is called an *epoch*. A time-ordered series of epochs is called an *era* (Ross and Rhodes, 2008). As a designer looks across many different epochs, there are some system designs that may consistently retain high utility and others that perform poorly in some epochs.

As the dynamic nature of the tradespace exploration becomes more explicit, the ‘ilities’, or time-dependent performance characteristics of the designs that become relevant when the context changes, become more important. One of these ‘ilities’ is flexibility. While evaluation of only the changeability of a design may be accomplished in a static tradespace (Ross et al., 2008), making judgments about the value of that changeability is subject to the designer’s prediction of the future. To identify which designs are valuably flexible requires analysis of how designs can be changed between epochs as well as static system performance, which goes beyond the changeability analysis.

Tradespace exploration also involves large amounts of data which grow with the number of time periods considered. Analyzing large data sets becomes increasingly difficult given computational constraints. In order to help designers identify the valuably flexible designs in a tradespace, Value Weighted Filtered Outdegree (VWFO) can be used to filter the tradespace for valuably flexible designs.



## **Chapter 4**

# **Case Application: Satellite Radar System**

To demonstrate Value Weighted Filtered Outdegree (VWFO), a case application Satellite Radar System (SRS) is presented. Past efforts by the U.S. military to field a space radar capability have failed in the design phase; canceled due to cost estimates in the tens of billions, key technology limitations, and opposing user preferences. Given the high profile and possible impact of this system, including flexibility may be a smart design choice. VWFO can inform trades for system flexibility, and identifies a unique set of designs from existing MATE metrics. This new subset gives designers another option as they trade performance, cost and risk.

The case application uses the RSC framework detailed in Chapter 3. Readers should note the extensive attention given to time-dependent context uncertainties in the problem formulation, as well as change mechanisms included in the Design Vector (DV). Six RSC processes are described as they pertain to SRS. VWFO is then applied to the resulting tradespace. The designs identified as being valuable flexible are called out and analyzed to confirm the heuristic results. General results

and limitation of VWFO are presented. The designs identified by VWFO as being valuably flexible are designs that do not generally appear on the utility-cost Pareto Front, and also don't have the maximum level of change mechanisms included.

## 4.1 Background on SRS

Radio detection and ranging, or radar, is the process of transmitting modulated waveforms using directive antennas to determine the range, velocity, and material composition of a target based on the analysis of variation in the reflected signal. Radars are all-weather active sensors, providing products during times when optical sensors are limited (Cantafio, 1989).

Radar was first used for military purposes by the British during World War II as a defense against air raids. Radar provided advanced warning of high flying aircraft. The critical nature of radar led to advances in technology in antennas and electronics. After the war, radar was turned to many diverse purposes including air traffic control, weather monitoring, and road speed control. Radar continues to be used by military services. The line-of-sight limitation of radar, as with most radio wave systems, has driven militaries to deploy radars at higher vantage points (Corcoran, 2000).

Previous studies from the 1990's examined moving airborne surveillance missions to orbital platforms, thereby gaining the ultimate high ground (DeLap, 1999; Wickert, 1997). Several of these studies found satellite masses were prohibitively expensive and recommended waiting for technology to mature, driven by commercial pull instead of government push. The advantages of a space-based system have caused the concept to be revisited periodically. Advantages, beyond gaining the high ground, include access to denied areas of the globe and continuous availability for near-real time tasking. Disadvantages of a SRS are primarily due to fundamental physics linked to technology limitations, leading to high costs. Range to target for an orbital platform will be hundreds of kilometers, while an airborne platform range is an order of magnitude less. The longer

range requires the transmitted power to be increased substantially due to returned signal attenuation. The radar range equation, ignoring losses is:

$$P_r = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 * R^4} \quad (4-1)$$

where

$P_r$  is the signal power received (W)

$P_t$  is the signal power transmitted (W)

$G$  is antenna gain (dB)

$\lambda$  is the wavelength of signal (m)

$\sigma$  is the target cross section  $m^2$

$R$  is the range m.

Other disadvantages include the inability of a space-based platform to loiter for long periods of time over targets. The range equation necessitates lower orbit altitudes to combat signal loss. Achieving continuous coverage of a target would require large numbers of satellites. Once in orbit, repairing and upgrading the satellites is nearly impossible except via software upgrades. This reduces the lifetime of the system as compared to comparable airborne platforms. The technology associated with space-based radar platforms is still being developed, and the end result of the technology development is still unknown (Davis, 2003).

## 4.2 Value Driving Context Definition

The designer, considering the diverse history of the system, identifies an appropriate stakeholder to begin the value elicitation. This process focuses on setting up the problem by eliciting a value proposition and future system uncertainties from enterprise stakeholders.

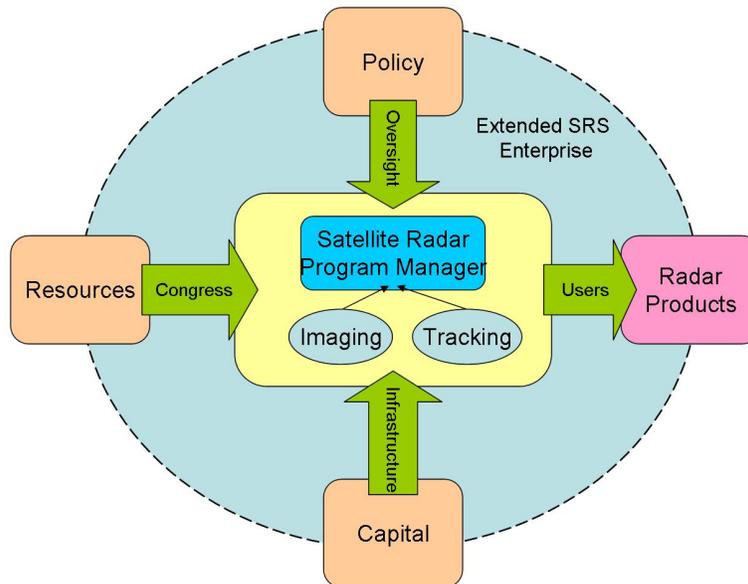
Initial inputs were taken from the stakeholder, the SRS ‘program manager’ (1.A.2). This stakeholder is in charge of developing, acquiring and fielding the system. The mission statement is

to build and deploy 24-hour, all-weather imaging and tracking capability (1.I.2). The value proposition is to evaluate candidate architectures over a large range of changing contexts (1.I.1). Two missions, Synthetic Aperture Radar (SAR) imaging and Ground Moving Target Indication (GMTI) tracking each have their own stakeholder, which is the person from whom the attributes and utilities were elicited. To resolve the multi-stakeholder problem, the program manager was appointed 'benevolent dictator' with preferences over the satisfaction of the mission stakeholders, forming a meta-utility function.

The SRS enterprise is defined as the system itself and consists of one or more spacecraft (1.A.1). This is a narrow engineering view. The extended enterprise includes stakeholders such as the eventual users of the system and the supporting elements required to complete the acquisition.

Exogenous stakeholders include Congress, which will approve budgets for SRS, and potential users of the system, which stand to gain slightly different capabilities dependent on the program manager preferences (1.A.4). The distance and disconnect between the end user, the agent acquiring the system, and the body of lawmakers budgeting the system is present for many military acquisition programs. The enterprise system boundary, shown in Figure 4-1, depicts the relationship between the program manager and endogenous and exogenous context factors. The program manager must consider internal influences like the mission stakeholders in addition to external factors like capital constraints and national policy (1.A.6). Previous MATE studies considered only factors inside the enterprise boundary. By considering influences crossing the enterprise boundary, more context uncertainties can be identified.

Epoch uncertainties were also identified and included technology development uncertainties for the spacecraft, budget constraints, operational target sets, and program manager preferences across the imaging and tracking missions (1.A.5). As interviews with the mission stakeholders proceeded, the value proposition was refined as follows: To determine which SRS architecture a notional program manager should select to maximize the chances that stakeholders will remain satisfied through the



**FIGURE 4-1:** Value Map for Satellite Radar System Enterprise

system lifecycle (1.A.7). Flexibility is one risk mitigation strategy that such a program manager should consider, and the metrics described in this research will operationalize the concept of flexibility into a tradable system property.

### 4.3 Value Driven Design Formulation

This process formalizes the general value propositions elicited previously into attributes and utility. The stakeholders are heavily involved in this process.

Stakeholder preferences are captured by interviewing the stakeholder and creating attributes (2.A.1). These attributes must be quantifiable and perceived independent. The SRS study has 12 attributes, shown in Table 4-1 (Ross et al., 2008). The attributes are across two different mission areas, imaging and tracking, and have two stakeholders corresponding to the missions. The aggregation method, Multi-Attribute Utility (Keeney and Raiffa, 1993), is only valid for a single

stakeholder, so the two missions were combined into a higher order “total utility” by the program manager who internally trades off preferences over these missions. Whenever utility is referred to in the case application, it refers to this total utility.

Each attribute has an associated single attribute utility function. This function ranges from zero to one, with zero utility as the least acceptable threshold of the attribute, and a utility of one indicating the attribute is fully satisfied. Designs that have zero utility for an attribute are considered the minimally acceptable, while designs with utility less than zero are not acceptable and deemed ‘invalid’. These design are eliminated as infeasible in the epoch tradespace.

In order to aggregate across the single attribute utility functions, shown in Figure 4-2 and Figure 4-3, each attribute is assigned a weight referred to as a ‘small k value’. The small k values for each attribute are listed in Table 4-2. The two mission areas are aggregated by the preferences of the program manager on missions, but this preference was selected as an epoch level uncertainty and will be discussed in the following section<sup>1</sup>.

A monostatic radar, or a system where the transmit and receive functions use the same antenna, was chosen as the system concept (2.A.2, 2.A.5). System concepts in this case application were limited for illustrative purposes. Availability of sufficient resources could allow for consideration of more than one system concept in the same tradespace.

The attributes tell the designer how the stakeholder will decide between designs. The designer takes the attributes and chooses aspects of the design that can be altered, and will drive those attributes. The potential design variables with first-order impacts are selected to be included in the

---

<sup>1</sup>The attributes are translated to a utility function using multi-attribute utility methods (Keeney and Raiffa, 1993). While formal methods exist for eliciting the mapping of attributes to utilities by interviewing stakeholders, these methods are resource-intensive and their specific output may be sensitive. In addition, multi-attribute utility methods were not part of the RSC development effort. Therefore, utility elicitation is simplified in this case application whereby the attribute set is based directly on interview data and the acceptability ranges and single attribute utility functions are based on order-of-magnitude estimates by the RSC development team.

**TABLE 4-1:** Attributes for Satellite Radar System

Attribute	Description
<b>Imaging Mission</b>	
Imaging Latency	The time between the imaging of a given target and when the full target image is downloaded to the ground
Field of View	The area of the earth that the Radar has access to within its normal range of motion
Geolocation Accuracy	The system's reported location of an image on the surface
Number of Targets Per Pass	The number of targets the Radar can image within a given target box for a single pass
Revisit Gap Time	The number of observations (i.e., passes) of a given target over the course of a single day
Resolution	The minimum separation between two targets that permits them to be distinguished by the Radar
<b>Tracking Mission</b>	
Tracking Latency	The time between the imaging of a given target and when the full target image is downloaded to the ground
Track Life	Length of time that a single target can be tracked (continually imaged)
Target Acquisition Time	The time interval between receiving a tasking order to observe a given location and actually acquiring the target as a function of gap time and target detection time
Min Detectable Velocity	Minimal velocity at which a target can be distinguished from the background
Number of Target Boxes	The number of target boxes (defined at a given size (Km <sup>2</sup> ) and consisting of targets with a given velocity and Radar Cross Section) that can be imaged by a single satellite during a single pass
Min Radar Cross Section	The minimal signal reflected from a target in response to a pulse that is capable of being detected by the radar's receiver

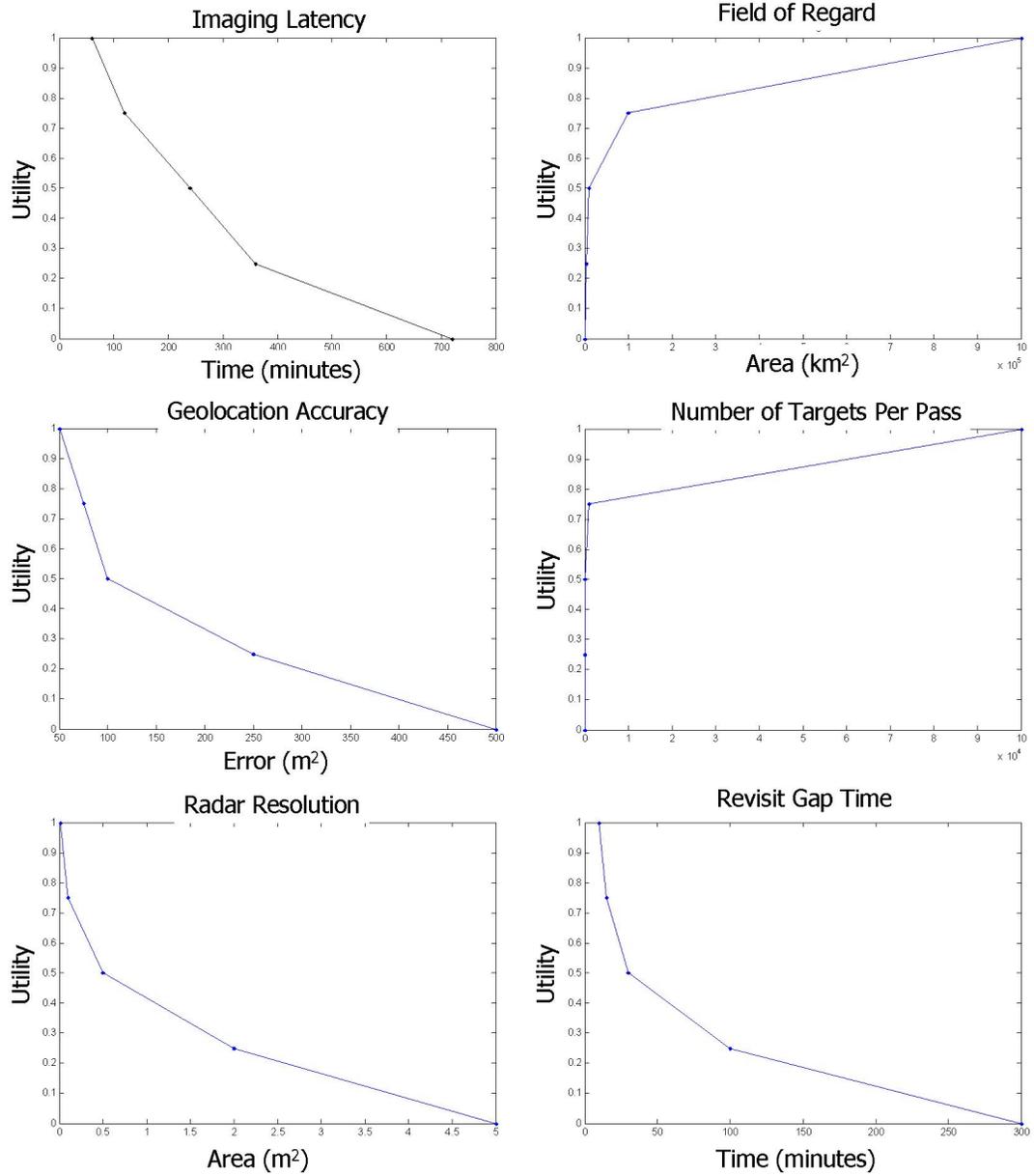


FIGURE 4-2: Utility Functions for Imaging Stakeholder

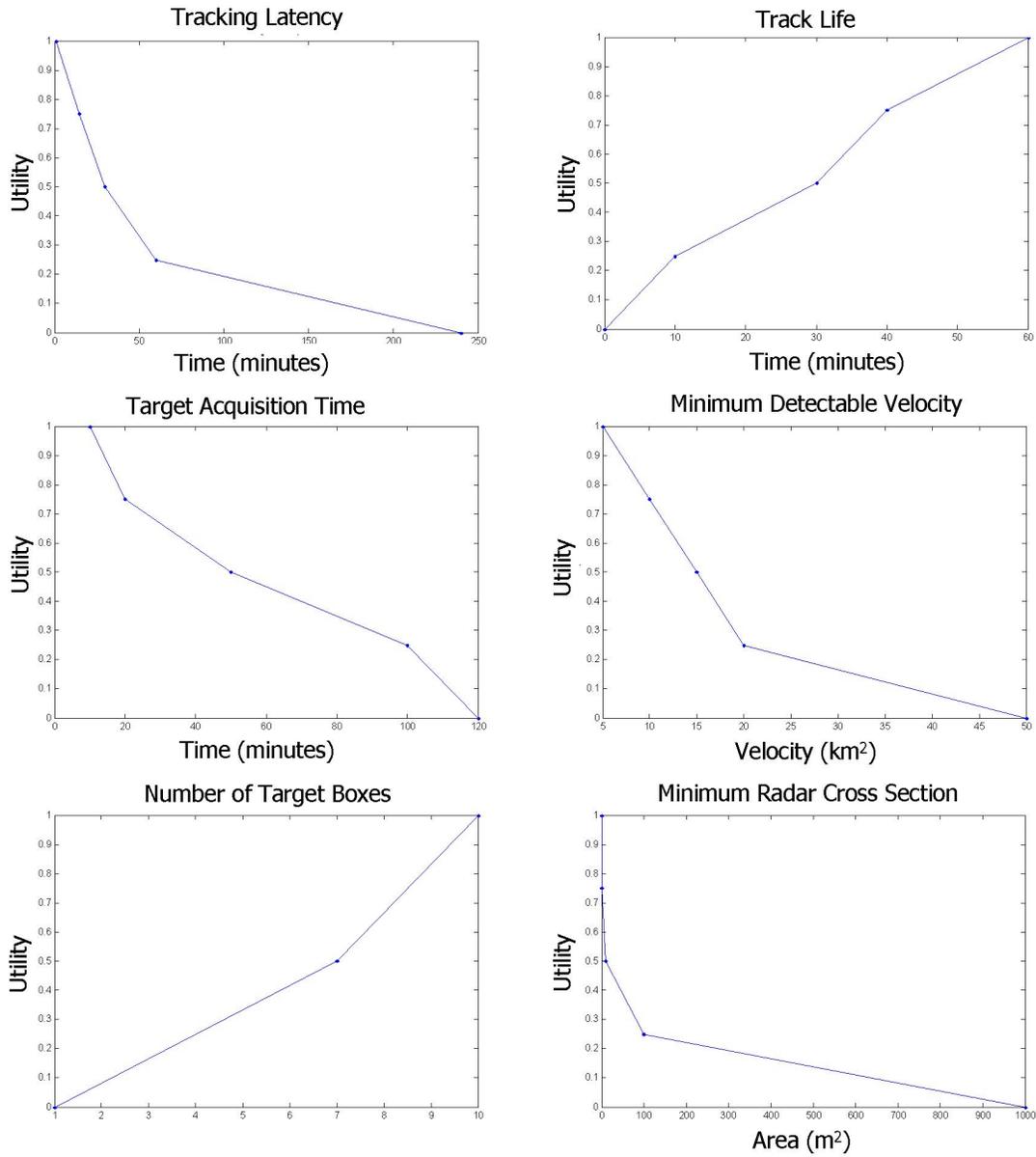


FIGURE 4-3: Utility Functions for Tracking Stakeholder

**TABLE 4-2:** Satellite Radar System Attribute Weights

Attribute	Weight
<b>Imaging Mission</b>	
Imaging Latency	0.04
Field of View	0.13
Self Geolocation	0.04
Number of Targets Per Pass	0.38
Revisit Gap Time	0.04
Resolution	0.38
<b>Tracking Mission</b>	
Tracking Latency	0.38
Track Life	0.13
Target Acquisition Time	0.04
Min Detectable Velocity	0.04
Number of Target Boxes	0.38
Min Radar Cross Section	0.04

model (2.A.4). This step of the process may be revisited during later iterations to check that designer intuitions were correct and all first order effects were captured in the design variable set.

The design variables chosen for this problem are shown in Table 4-3 (Ross et al., 2008). The goal of the design variable enumeration was to cover a large area of the design space for the concept defined above. The parametrization of the design variables is called the Design Vector (DV) (2.O.5). A design is considered a unique set of design variables in a DV.

The design variables are in three categories: radar technology, orbit/constellation design, and vehicle change mechanisms. A full factorial enumeration of this design space leads to 23,328 individual designs. The Design Value Map (DVM) in Figure 4-4 relates the anticipated impact of the design variables to the attributes (2.O.3). This construct does not attempt to evaluate the DV, nor determine the utility of the designs. The DVM acts as a first order check to see if the design variables will drive the attributes and helps the designers begin thinking about the model framework. This step may cause a feedback loop if the designer realizes that the attributes are not computable within the defined enterprise boundary.

**TABLE 4-3:** Design Variables for Satellite Radar System Case Application

Design Variable	Description
Peak Transmit Power (1.5 10 20 [kW])	The amount of power that is used to send the radar signal to illuminate the target area. The higher the power sent, the higher the return signal, in general
Radar Bandwidth (0.5 1 2 [GHz])	The bandwidth of the radar signal. Larger bandwidths generally return better signatures, but have less power spectrum and so require more power
Antenna Area (10 40 100 [ $m^2$ ])	The size of the AESA array.
Altitude (800 1200 1500 [km])	The altitude of the satellite in orbit
Constellation Design (8 Walker IDs <sup>a</sup> )	The Walker configuration that is chosen for the spacecraft
Comm Downlink (Relay or Downlink)	Whether the satellite has a communications system able to use a dedicated communications backbone, such as TDRSS
Tactical Downlink (Yes or No)	Whether the satellite is designed with a high power, localized downlink for tactical users
Maneuver Package (1x, 2x, 4x)	The total amount of maneuvering fuel on-board the satellite
Constellation Option (none, long-lead, spare)	What real option is built into the supply chain

<sup>a</sup> Walker ID is a lookup table of characteristics for a Walker constellation, with different possibilities for number of satellites and the phase(plane orientation). The Walker constellations are shown in Table 4-4.

TABLE 4-4: Walker Constellations

Walker ID	Inclination (degrees)	Number of Satellites	Number of Planes
1	53	5	5
2	53	10	10
3	53	10	5
4	53	20	5
5	67	5	5
6	67	10	10
7	67	10	5
8	67	20	5

Design Vector		Attributes														
		Tracking							Imaging							
		Minimum Target Radar Crosssection	Minimal Discernable velocity (ref. CS)	Number of Tgt Boxes	Target ID Time = f(gap t, detection t)	Target Track Life	Tracking Latency	Resolution (Proxy)	Targets per pass	Field of Regard	Revisit Frequency	Imaging Latency	Geolocation Accuracy	Actual Costs (Era)	Total Impact	
Variable Name	Definition Range															
Peak Transmit Power	5 - 15 KW	9	9	9	3	1	1	9	9	9	0	1	3	9	72	
Radar Bandwidth	TBD (MHz)	9	9	3	3	1	1	9	9	9	0	1	3	3	60	
Physical Antenna Area	10 - 100 m <sup>2</sup>	9	9	9	3	1	1	9	9	9	1	1	3	9	73	
Antenna Type	Mechanical vs. AESA	9	9	9	3	3	1	9	9	9	1	1	1	9	73	
Satellite Altitude	700 - 1500km	9	9	3	9	9	3	9	9	9	9	3	9	1	91	
Constellation Design (list)	Walker ID	0	0	1	9	9	3	0	0	3	9	3	9	9	55	
Comm. Downlink	Sat Backbone vs. Downlink	0	0	0	0	0	9	0	0	0	0	9	0	9	27	
Tac. Downlink	Yes vs. No	0	0	0	0	3	9	0	0	0	0	9	0	9	30	
Maneuver package	1 - 6	1	1	1	1	1	0	1	1	1	1	0	1	3	13	
Constellation Option	1-3															
Total		46	46	35	31	28	28	46	46	49	21	28	29	61		

FIGURE 4-4: Design Value Map relating Design Variables with Attributes

**TABLE 4-5:** Epoch Variables for Satellite Radar System Case Application

Epoch Category	Epoch Variables	Levels
National Security Policy	Notional Target Sets User Priority (Image vs Track)	9 scenarios SAR = GMTI, SAR < GMTI, SAR > GMTI
Technology	Available Technology Levels	High, Low
Infrastructure	Communications Infrastructure	2 Levels
Systems-of-Systems	Airborne ISR Assets	3 Levels
Environment	Communications Jamming	none, 10dB

#### 4.4 Epoch Characterization

While the previous process dealt almost exclusively in static thinking about the system, this process will consider the dynamic aspects of the environment and user preference changes on the system.

The epochs were defined using several epoch variables (Roberts et al., 2009). These variables describe the context: both the environment in which the system operates, and the stakeholder preferences (3.A.1). Like for the computer design model, inputs to the future scenarios were modeled with discrete enumeration levels. Again, this step was limited by time constraints as well as memory constraints. Analyzing a rapidly growing data set posed an additional computational challenge. Another DVM with the proposed list of epoch variables was created, and the epoch variables with first order impacts were selected for inclusion in the model (3.A.4).

The Epoch Vector (EV) is defined in Table 4-5. These uncertainties were parameterized like the DV in the previous step. A full enumeration of the EV results in 648 epochs (3.A.6) of which 245 were simulated.

Given the large set of possible contexts, the decision was made to reduce the epoch space by



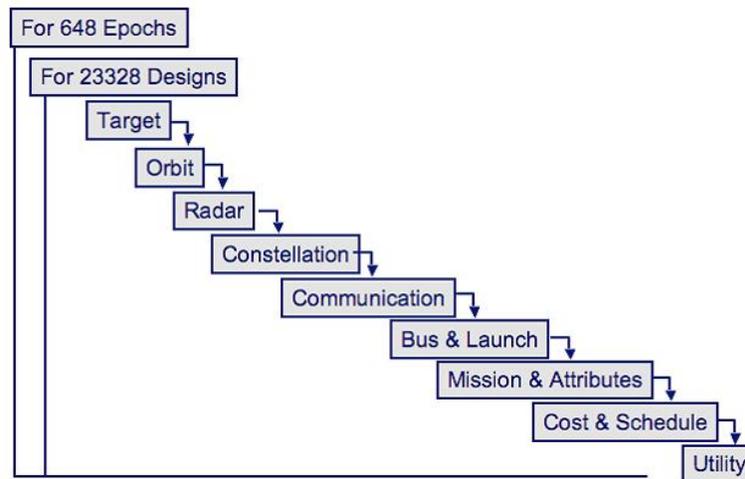
grouping some of the above changes together to obtain the initial epoch vector, as shown in Figure 4-5. The enumeration levels of the epoch variables in the vector were also chosen in order to keep the computation time to a level reasonable given the constrained computational resources available for this project. However, the coarse sampling was done in a way that still captured the dynamic properties of the system over many diverse context changes.

The epoch variables will now be briefly described. Each target set consisted of two targets, specified by notional country (actually quantified by latitude) and target RCS. For the epoch, attributes are computed against the worst case of the two targets. When calculating overall utility, the utilities calculated for the two users considered are weighted in 3 different ways, indicating the program manager's preference over the two mission areas. Technology level includes radar and bus technology changes, with high and low levels considered (see Figure 4-5). "Communications Infrastructure" levels indicate the availability of ground stations and space-based relay options; level one has no relay but can use AFSCN stations, and level two has WGS and AFSCN stations. AISR levels indicate the availability or non-availability of two different airborne assets. "Environment" indicates communications with jamming or without jamming.

## **4.5 Design Tradespace Evaluation**

Now that the designer has a list of attributes, DVs, and EVs, the computer model that will translate designs into utility is implemented. The SRS model is depicted in Figure 4-6. The modules were based on function area of the spacecraft technology, orbital physics and translating performance in attributes to utility. Cost estimation for the design was also conducted (4.A.1).

Each module of the code is briefly described below to illustrate the logical flow of the model (4.A.2). Modules early in the flow take in DVs and modules later in the flow determine support structure for the spacecraft which are not necessarily determined by the DV.



**FIGURE 4-6:** Module Framework for SRS Computer Module.

*Target*- Defines worst-case target from the epoch variable target sets.

*Orbit*- Calculates the properties of the orbit that affect satellite performance, assuming spherical Earth, circular orbits, cylindrical shadow cast by the earth, and constant satellite velocity.

*Radar*- Computes the performance of a radar specified by the DV, in the orbit calculated above, looking at the target calculated above. Assuming the spacecraft will operate to maximize attributes; basic physics may be used to calculate radar performance.

*Constellation*- Simulates the operation of the constellation of SRS vehicles operating over a turning earth with a set of targets, users, and communications infrastructure specified by the epoch. They are used to calculate coverage attributes and provide input into communication calculations.

*Communication*- Estimates data latency and data throughput attributes as well as the cost, mass, and power consumption of the communication system.

*Bus*- Calculates the characteristics of the spacecraft necessary to support the radar and communication gear, including calculations with some detail of the structure (particularly the radar

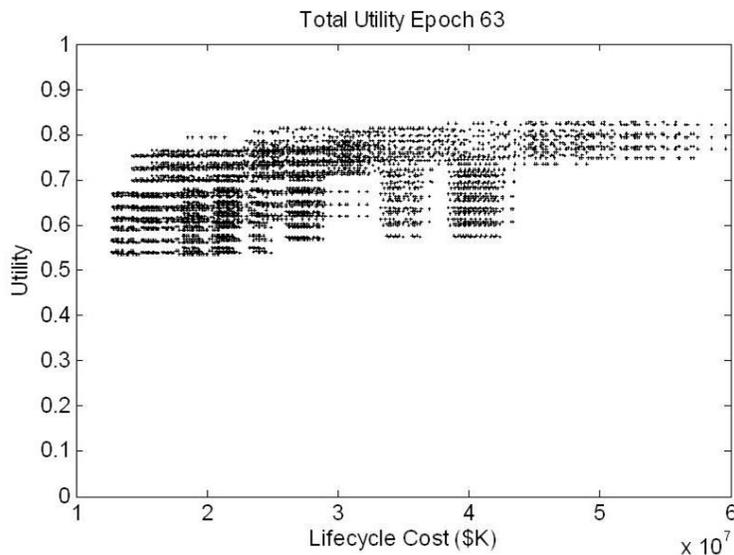
support structure), power, and propulsion systems, and design rule calculation of other systems (ADCS, thermal control, bus structure and adapters, etc.). It outputs the mass and cost of the vehicles and determines the cost of the least expensive US vehicle(s) and US site required to launch them.

*Mission-* The mission module is responsible for taking the outputs from the previous modules and calculating the attributes. These attributes are then fed into the utility module.

*Cost and Schedule-* Calculates the cost of the design and estimates a schedule for acquisition based on probable staffing.

*Utility-* Translates the attribute performance into single attribute utility values and applies multi-attribute utility to aggregate over each mission area.

For each epoch considered, the model calculates the expected utility for each valid design. A sample tradespace produced by the model is shown in Figure 4-7 (4.A.3).



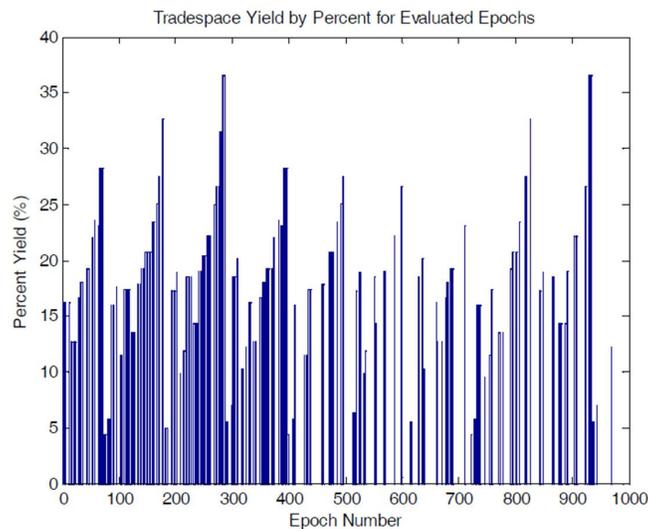
**FIGURE 4-7:** Tradespace With 5400 Valid Designs

The tradespace shown in Figure 4-7 is representative of the tradespaces calculated in the study. This is Epoch 63, which has the following EV: [Target Set 60, SAR>GMTI, Low tech level, No backbone, No AISR, No jamming]. Of the 23,328 designs calculated in this epoch, only 5400 met all minimum utility thresholds and were physically feasible. The Pareto Front is flat except in the low cost extreme of the range. The less expensive designs tend to be those that are invalid or just meet utility thresholds. The costs range from \$10B to \$60B. This lifecycle cost estimate is for an entire constellation including launch costs for ten years. The flat tradespace indicates that it is feasible to achieve high utility for a given cost budget (4.A.4, 4.A.5).

In this part of the RSC process, the designer's intuition about the impact of a design variable on attributes is assessed. For instance, it was found that the frequency used for the radar transmission had little effect overall on the attributes, so it was dropped as a design variable in the final version of the model. Extensive analysis in this process of RSC will reveal designs that deliver high value and will satisfy the stakeholders assuming that the context remains fixed. As this assumption is rarely true, the designs must be examined over many epochs using Multi-Epoch Analysis.

## 4.6 Multi-Epoch Analysis

This process analyzes designs during many different epochs. In general, insights from static tradespaces gained during the previous process step (e.g. the trades between attributes and value drivers), will hold true across all epochs. This step is looking for the design drivers across epochs. A first look at how the changing epochs are affecting the design space is to consider Tradespace Yield. Figure 4-8 shows the results for all the evaluated epochs. Some epochs were not evaluated due to computational limitations leading to time constraints and are shown as a blank space in the figure.



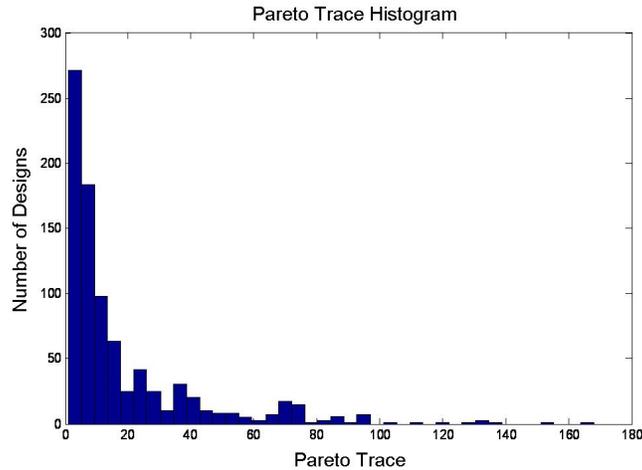
**FIGURE 4-8:** Tradespace Yield for Evaluated Epochs

The Tradespace Yields vary between 4% and 36% of the total designs enumerated (5.A.2c). The epochs with very low yield tend to have very challenging target scenarios. Very high yield epochs tend to have higher technology and Airborne Intelligence Surveillance and Reconnaissance (AISR) assets. The variance between epochs highlights the importance of considering context changes during conceptual design.

#### 4.6.1 High Pareto Trace Designs

The Pareto Trace of a design indicates how passively value robust that design is (Ross, 2006; Ross et al., 2008, 2009). *Passively value robust* describes a design that, without changing, retains high stakeholder utility over many contexts. Figure 4-9 is a histogram that shows that there are many designs that appear in a few Pareto Sets. Those designs which have high Normalized Pareto Trace are found in Table 4-6 (5.A.2).

In this case study, the Pareto set includes the designs that are highest in total utility for a given cost.



**FIGURE 4-9:** Histogram of Pareto Trace for 245 Epochs

**TABLE 4-6:** Utility for High Normalized Pareto Trace Designs

Design Number	Normalized Pareto Trace	Total Utility			
		Epoch 63	Epoch 171	Epoch 193	Epoch 202
3435	0.69	0.75	0.75	0.57	0.66
3447	0.55	0.76	0.76	0.63	0.73
3555	0.56	0.82	0.82	0.81	0.90
6027	0.62	0.76	0.76	0.55	0.65
6039	0.53	0.76	0.76	0.61	0.71
6147	0.53	0.81	0.81	0.81	0.90

Pareto Trace for the design is normalized against the total number of epochs. Therefore, the Normalized Pareto Trace (NPT) is a number from zero to one, with one indicating designs in all epoch Pareto Sets and zero indicating a design that never appears on the Pareto Front of the epochs that have been sampled. A first look at this set of data reveals that there is one design with the highest Pareto Trace: design number 3435, with an NPT of 0.69. Designs with NPT greater than 0.5 are shown in Table 4-6 (5.A.3).

### 4.6.2 Filtered Outdegree

Filtered Outdegree (FOD) is a metric that identifies changeable designs (Ross et al., 2008). Unlike Pareto Trace, Filtered Outdegree does not depend on the utility of the system. For this case application, a single transition rule was chosen for demonstration, called “redesign”, which enables the change of a design variable from one enumerated level to another, while incurring an associated transition cost, increasing as the cost of the system increases (5.1.2). In this instance, the costs are captured by money and time. Some transitions may involve changing the SRS orbit configuration, and these rules often impose a fuel cost as well. FOD measures the number of designs that a particular design may transition to at acceptable cost. However, since FOD in no way implies anything about the utility associated with those changes, it is difficult to say how valuably flexible something is based only on the number of allowable transitions (5.A.1).

The details of how to calculate FOD for this study are as follows. Each design that is calculated in the tradespace has a design vector, a performance vector (attributes with utility), associated constants, calculated intermediate variables (e.g. mass), and an epoch vector. The FOD for each design is calculated for each epoch (i.e., we assume that the context remains the same when calculating costs). In order to be “counted”, the performance of a design must meet the minimum utility specification for the epoch.

A transition path represents a possible change from one design to another. For instance, design 3435 has a 40 meter antenna. If the size of the antenna were to be changed to 100 meters instead, that would result in a path between design 3435 and 3543, the design that is exactly the same as 3435 except for the antenna size. The path in the tradespace is the arc that is notated between these two design points. All paths for a given design are assumed to be directional to the other design. The destination design may then have an opposite direction path that brings it back to the original design. These two paths or, ‘arcs’, are considered distinct and counted separately.

**TABLE 4-7:** Utility for High Filtered Outdegree Designs

Design Number	FOD Filter is \$10 <sup>7</sup>	Total Utility			
		Epoch 63	Epoch 171	Epoch 193	Epoch 202
1089	188	0.62	0.63	0.49	0.60
1101	359	0.72	0.72	0.55	0.66
8921	488	0.77	0.78	0.65	0.75
9029	278	0.77	0.78	0.81	0.91
16701	437	0.77	0.78	0.65	0.75
16809	263	0.77	0.78	0.81	0.91

The existence of an arc indicates that a transition is possible. Once the arc is established, it needs to have an associated cost and time. Most transitions are costly in terms of both time and money. Changing designs will incur a transition cost, which includes a ‘friction’ cost representing inefficiencies and effort in executing a change, and this will apply even if the design moves from a very expensive design to a less expensive one. Anticipated cost savings do not materialize in full because of the associated friction cost. The transition cost is the notional cost to the program to change from one design to another. A designer then designates a transition cost, above which he would be unwilling to incur the expense of changing the system design. This threshold is then applied to all transitions, and only those less than the threshold, or ‘filter’ are allowed. The designs shown in Table 4-7 were found to have the highest filtered outdegree (5.A.4).

## 4.7 Era Construction

To construct eras, scenario planning was utilized. Several epochs are chosen in sequence based on domain expert knowledge. These comprise a small number of the calculated epochs, leading to a small set of eras. One era was chosen for demonstration in this case application and is shown in Table 4-8 (6.A.1).

The era describes a simple scenario in which the system progresses through changes to the

**TABLE 4-8:** Description of Epochs in Era 1

Epoch Number	Duration (yrs)	Description	Epoch Vector
63	2	Today	Scenario 60, SAR > GMTI, Low tech level, No backbone, No AISR, No jamming
171	4	Natural advancement in Comm capability	Scenario 60, SAR > GMTI, Low tech level, Backbone, No AISR, No jamming
193	1	New threat- Mobile missile launchers	Scenario 94, SAR > GMTI, Low tech level, Backbone, No AISR, No jamming
202	3	Increased Conflict, UAVs deployed, Jamming	Scenario 94, SAR < GMTI, Low tech level, Backbone, AISR, Jamming
171	3	Conflict Resolved, back to pre-conflict state	Scenario 60, SAR > GMTI, Low tech level, Backbone, No AISR, No jamming

stakeholder preferences and target set requirements (Roberts et al., 2009). This era has five epochs, four of which are unique. Epoch 63 is the first, then the era progresses to Epoch 171, which has a different target set. The stakeholder preferences remain the same, while the performance of the systems changes slightly. Next in Epoch 193 the same target set exists, but the stakeholder preferences have shifted, causing the utility of the designs to change dramatically. Epoch 202 retains the same stakeholder preferences, but switches the target set. The era concludes with the stakeholder preferences and target set returning to the conditions found in Epoch 171.

#### 4.7.1 Value Weighted Filtered Outdegree

Now that the era has been identified, it is time to apply VWFO to the tradespace shifts. The high passive value designs (those that retain value without changing), and the highly changeable designs (those that have many transitions available), were identified in earlier steps using NPT and FOD.

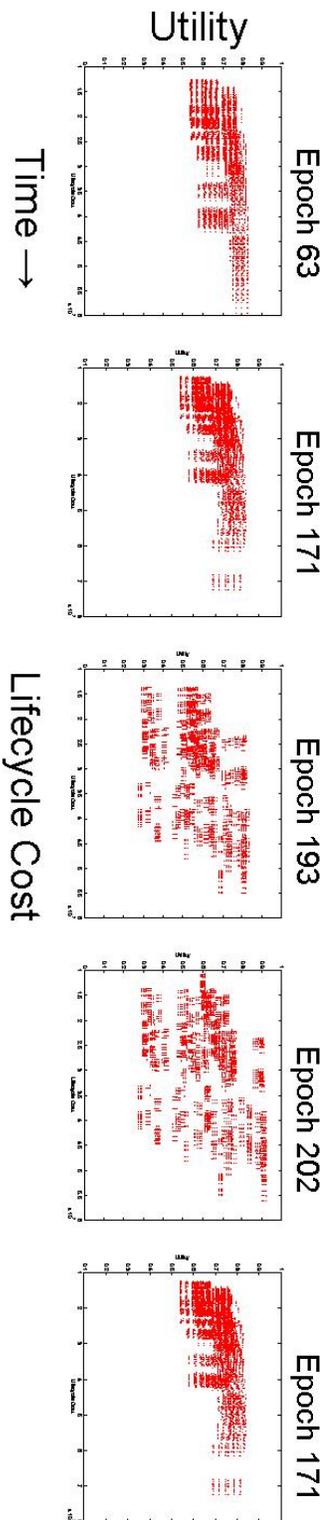


FIGURE 4-10: Tradespace Storyboard of Era 1

To bridge the gap and find the designs that are valuably flexible, the VWFO is assessed for all designs in the tradespace. The designs that have some useful transitions and relatively high utility are the designs that can be considered valuably flexible. The metric to identify these designs is called *Value Weighted Filtered Outdegree*.

$$VWFO_i^k = \frac{1}{N-1} \sum_{j=1}^{N-1} [\text{sign}(u_j^{k+1} - u_i^{k+1}) * \text{Arc}_{i,j}^k]$$

where

$N$  is the number of designs considered

$k$  is the current epoch

$k + 1$  is the next epoch in the era

$i$  is the design under consideration

$j$  is the design to be transitioned to

$u_i^{k+1}$  is the utility of design  $i$  in the  $k + 1$  epoch

$u_j^{k+1}$  is the utility of design  $j$  in the  $k + 1$  epoch

$\text{Arc}_{i,j}^k$  is the logical value indicating the presence of a transition arc from design  $i$  to design  $j$

This metric captures the utility difference in the destination designs and is dependent on how many transitions are available. The intent of the metric is to act as a screening heuristic, and it is left to the decision maker to make a final call on the value of the design.

DVs for all designs are shown in Table 4-12 at the end of the chapter.

The VWFO for twelve example designs discussed in previous sections is shown in Table 4-9.

Some designs have zero VWFO. There are several reasons this may be the case: the design may have zero transitions available, the transitions available result in zero net utility, or the design may not be available in the  $k + 1$  epoch. Designs with positive VWFO (e.g. 1098, 1101, 3435 and 3447) have transitions available to higher utility designs in the first epoch change. This may indicate that these designs are valuably flexible. In a previous section, design 3435 was identified as the highest

**TABLE 4-9:** Value Weighted Filtered Outdegree for Selected Designs with high Filtered Outdegree or high Pareto Trace

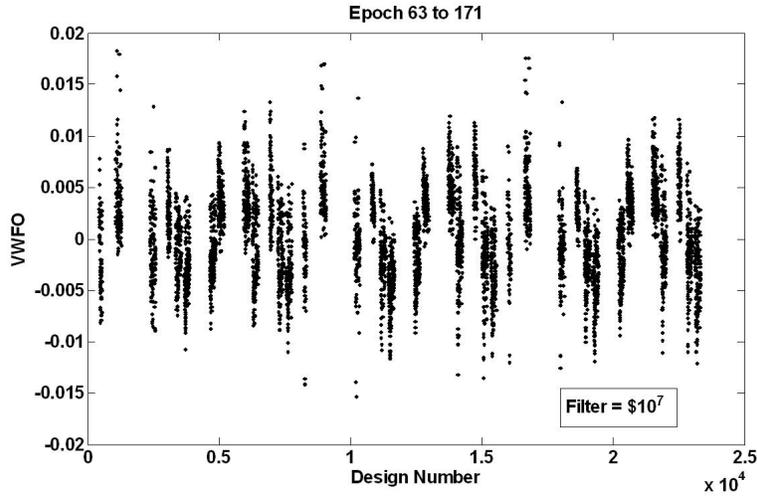
Design Number	Value Weighted Filtered Outdegree			
	63 to 171	171 to 193	193 to 202	202 to 171
1089	0.0044	0.0048	0.0000	0.0000
1101	0.0085	0.0097	0.0021	0.0032
3435	-0.0057	-0.0021	-0.0034	-0.0031
3447	-0.0035	-0.0017	-0.0031	-0.0027
3555	-0.0014	-0.0006	-0.0014	-0.0014
6027	0.0044	0.0054	0.0020	0.0024
6039	0.0027	0.0031	0.0023	0.0027
6147	0.0019	0.0022	0.0014	0.0017
8921	0.0147	0.0187	0.0064	0.0072
9029	0.0000	0.0388	0.0077	0.0091
16701	0.0089	0.0138	0.0043	0.0051
16809	0.0000	0.0138	0.0059	0.0067

NPT design. As the Era progresses, the VWFO for the designs change because the utility of the designs changes across the epochs.

One way a decision maker can use this information is to look at several eras. Additionally, it may be that the designs with high VWFO share the same change mechanisms. This would indicate that this particular change mechanism is exercised frequently, for instance, varying the amount of fuel carried on board the satellite.

Designs with high magnitudes of VWFO may be more valuably flexible than others. Designs that have positive VWFO are able to transition to destination designs that have higher net utility.

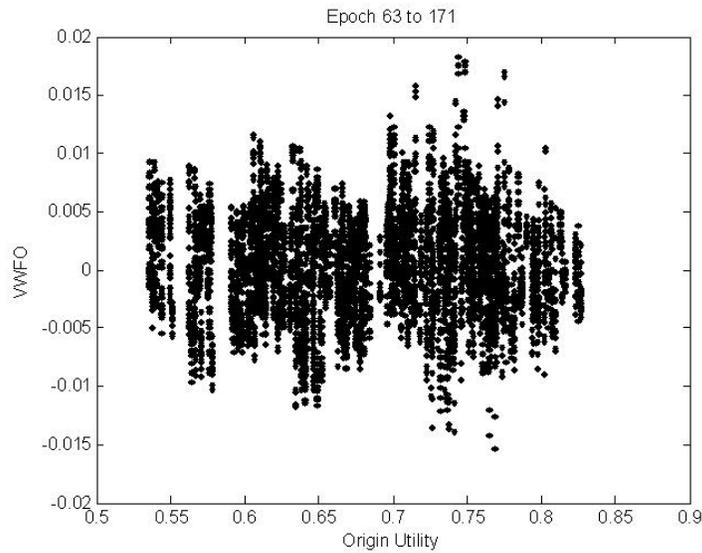
Unlike choosing designs based solely on high NPT or high FOD, VWFO can identify designs that are valuable and flexible. VWFO takes into account the value of the change (the utility change direction), the changeability of the design (transition arcs), and the context changes (era progression).



**FIGURE 4-11:** Value Weighted Filtered Outdegree for Epoch 63-171

Figure 4-11 plots the VWFO for an entire tradespace by design number. The striations in the space are caused by the discrete enumeration of the design space. (A plot highlighting high NPT and high FOD designs is shown in Figure 4-15.)

To make sure that the metric was not biased by any unfavorable relationships with other variables, several checks were conducted. The first compared the utility of the origin design (design  $i$ ) against the resulting VWFO for that design. Figure 4-12 shows the spread of points. The designs are fairly evenly distributed against the area, and no correlation is seen. The  $R^2$  value for the correlation of origin utility to VWFO is  $-0.0428$  with a  $p$  value of  $5 \cdot 10^{-4}$ , indicating there is no linear correlation between these two. Due to the design space discrete enumeration, the correlation between design number and VWFO is slightly higher, at  $0.0143$  with a  $p$  value of  $0.2531$ . There is still no significant linear correlation, although the slight increase in  $p$  reflects the striations noticed in Figure 4-11.



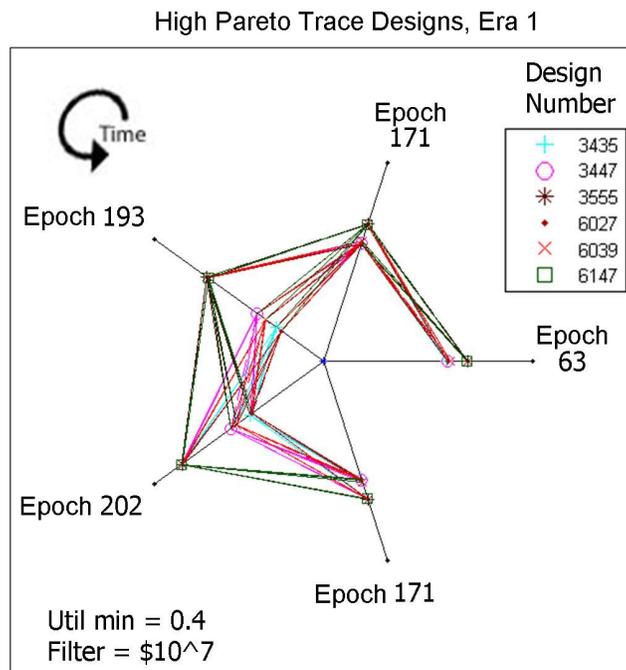
**FIGURE 4-12:** Total Utility of Origin Design with VWFO to Check for Linear Correlation

### 4.7.2 Era Spider Plot

The amount of information available to a designer in a tradespace is very large. Typically, a designer will run analysis on the tradespace as a whole and determine several ‘interesting’ designs for deeper analysis. These subsets of designs can further increase the designer’s knowledge of possible trades. Visualizing this information in a dynamic way is difficult. Because flexibility is dependent on the context change to reveal value to the decision maker, a way to visualize the designs over time is required.

One way to look for flexibility is with a spider radar plot. Inspired by the Performance Gap metric introduced by Mark (2005) the spider plot contains information about the performance of the SRS system, as well as the possible transitions.

In Figure 4-13, the epochs occur sequentially counterclockwise, beginning with Epoch 63. As time progresses, the radials of the spider plot represent a snapshot of each epoch. The ends of the radials



**FIGURE 4-13:** Spider Plot Containing Six Designs with high Pareto Trace

indicate a stakeholder utility of one while the center of the radials is set at 0.4 to scale the plot for clarity. The stringers (lines between design points) between epochs represent transition arcs. The presence of a stringer indicates that a transition is possible, at a cost less than a threshold cost filter, which is noted in the lower left corner of Figure 4-13.

Dashed stringers are used to help distinguish designs that ‘stay the same’ across consecutive epochs. These are not transition paths, however ‘do nothing’ is a valid strategic decision for a stakeholder experiencing epoch changes, and should be considered explicitly. These dashed lines can be used to identify the progression of static designs across the era.

The six designs that have high NPT in Table 4-6 were evaluated for several epochs in the era and were then plotted on a spider plot representation as described above. Only transitions to designs in the subset were allowed. This essentially takes a tradespace of 23,328 designs and reduces it to a

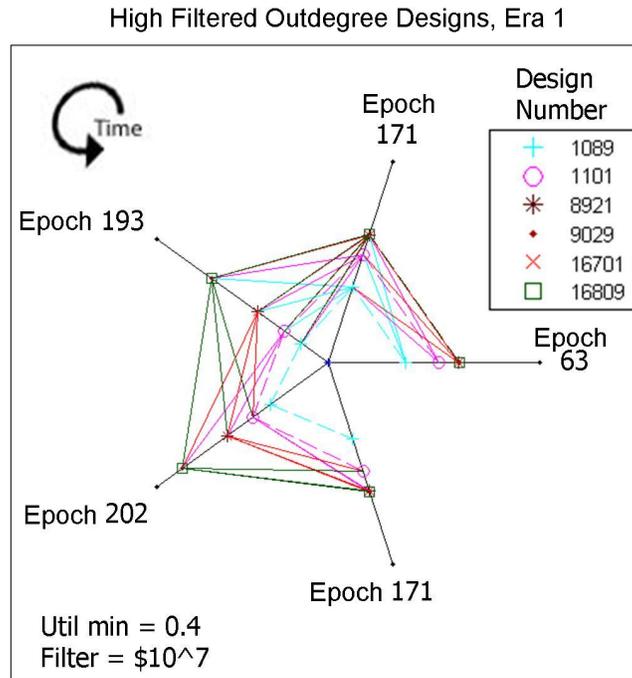
tradespace of six. Figure 4-13 shows the results for ‘redesign’ transition rule in the design phase with a cost threshold filter of  $10^7$  dollars. This filter represents an acceptable threshold for a transition cost less than the cost of the system.

When the high NPT designs are plotted they all have similar utility. In Epoch 63 the highest utility design, 6147, has a transition available to design 3555 in the next epoch, where that design is also high utility. The context change from epoch 171 to 193 is more significant. In epoch 193, some designs drop utility significantly compared to others. For instance, design 6027 has the lowest utility in Epoch 193, but is not the lowest in Epoch 171. There are a few transitions from lower utility designs to the higher utility designs. The designs that have change mechanisms available, and do not continue to provide value are not robust or valuably flexible. Other designs are able to recover from utility loss and transition to a higher utility design in the next epoch. Design 3447, which has lower utility in Epoch 193, has a transition available to design 3555, which in Epoch 202 has higher utility than design 3447. In this way, a decision maker can analyze designs and design interactions to determine if there are valuably flexible designs in the subset<sup>2</sup>. Note that this analysis used a single transitions rule called ‘redesign’ which is most appropriately used in the pre-ops phase of the satellite. A detailed analysis should use transition rules that apply to the current lifecycle phase of the system.

To compare the passively value robust designs (represented by the high NPT designs) to highly changeable designs (represented by high FOD designs), the second subset of designs is plotted in Figure 4-14. The differences between the two subsets of designs are extremely apparent in this representation. The high NPT designs have higher utility, but fewer possible transitions than high FOD designs. Again, the transitions available are only those within the subset. An interesting aspect of this subset of designs is the lack of reordering in the utility scale. Unlike the high NPT

---

<sup>2</sup>There are limitations to looking at a subset of designs instead of the tradespace as a whole. Filtered Outdegree is relative to the designs included in analysis. If a subset of designs is chosen such that the number of possible transitions is very small, the designs may not appear flexible at all. As a designer starts to look in detail at small regions of the tradespace, it is important that the big picture remains present in the analysis.



**FIGURE 4-14:** Spider Plot Containing Six Designs with high Filtered Outdegree

designs, which tend to be designs up against constraints, the highly changeable designs are not Pareto Optimal and meet the utility requirements without encroaching on boundaries. This means that as the era progresses, the decision maker may perceive that the best strategy is to choose the highest utility design of the subset and then remain with the same design. Not explicit in this representation is the cost of the designs themselves. These designs, not being Pareto Optimal, are dominated in utility space by less costly designs. However, the designs in this subset are highly connected by transitions, and in the event that the future unfolds in a different era, the value of being able to maintain utility over time is more likely, offsetting the apparent cost and “dominance” in the static view.

When the small subset of designs are shown within the entire tradespace (Figure 4-15) there are

several designs that have higher VWFO than the subset. These designs tend to be in the same region of the tradespace as the high FOD designs, but not in the same region as the high NPT designs. Several of the designs that were identified as having high FOD do not have high VWFO. These designs, while highly changeable, do not have many transitions to higher utility designs, meaning they are less flexible in this epoch transition.

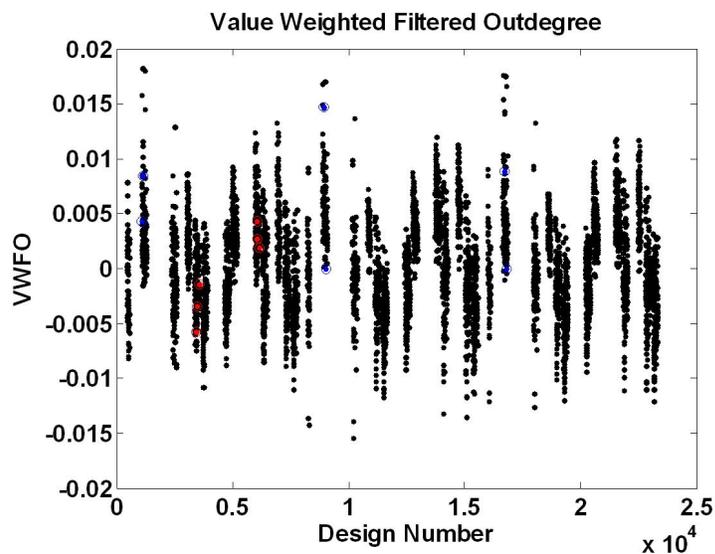


FIGURE 4-15: Selected Designs in Context of Entire Tradespace

Several designs with high VWFO are listed in Table 4-10. Design 1109, identified as having the highest VWFO is the transition from Epoch 63 to Epoch 171, also has the highest VWFO for all the epoch transitions. As the era progresses, the designs retain high VWFO over the epoch transitions. This may be caused by the similar epochs in the era. If the epochs were drastically different, the result may differ as well. Looking at an expanded set of eras, and the design behavior across the epochs, is a matter for future analysis.

When the designs in Table 4-10 are placed on an era spider plot, three groups of designs can be identified based on their utility. Note that some of these high VWFO designs were also identified as having high FOD. However, there are no designs with high NPT, those are found in a different

**TABLE 4-10:** Design with High Value Weighted Filtered Outdegree

Design Number	Value Weighted Filtered Outdegree			
	63 to 171	171 to 193	193 to 202	202 to 171
1097	0.0159	0.0166	0.0054	0.0054
1109	0.0183	0.0228	0.0079	0.0080
1205	0.0180	0.0213	0.0077	0.0074
8873	0.0149	0.0125	0.0038	0.0049
8885	0.0168	0.0165	0.0053	0.0072
8921	0.0147	0.0187	0.0064	0.0072
8981	0.0171	0.0177	0.0056	0.0068
9017	0.0171	0.0193	0.0066	0.0072
16649	0.0154	0.0138	0.0044	0.0052
16661	0.0176	0.0188	0.0063	0.0076
16757	0.0176	0.0192	0.0064	0.0071
16793	0.0166	0.0208	0.0074	0.0070

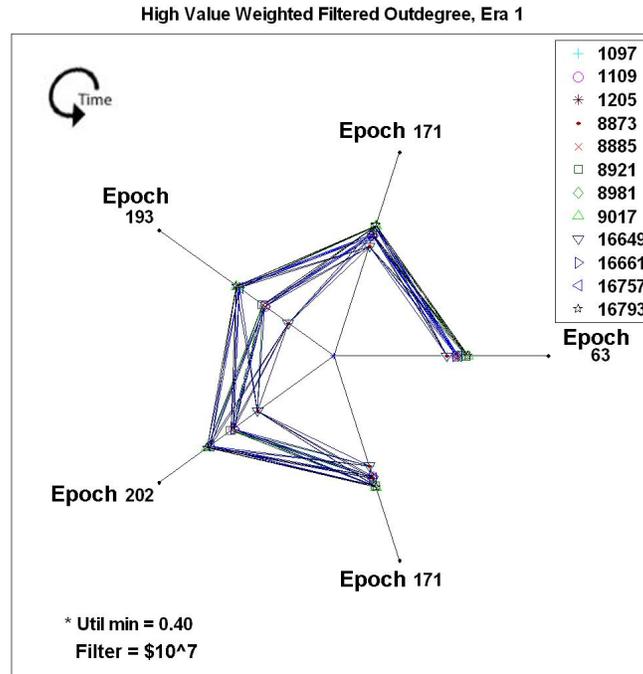
area of the tradespace. This spider plot appears similar to the high FOD plot, as there are many transitions identified and the designs are of similar utility. However, one of the drawbacks of the era spider plot is the limited number of designs that may be shown on such a plot. Figure 4-16 shows 12 designs, which is 0.2% of the valid designs and only 0.05% of the total tradespace.

## 4.8 Impact on Design of Adding Flexibility

Adding flexibility to designs is not free, as the carrying costs for change mechanisms are paid for up front. The impact of adding options to SRS is clearly apparent in Table 4-11. The table lists three families of designs. The first third is based on design 3435, chosen because it is the highest NPT design. The second section is based on design 8921, the design with highest FOD. The last section is based on design 1109, a design with high VWFO. The other designs have the same DV, except for the Maneuver Package and Constellation Option. These two design variables do not add utility to the design in a static context, but increase the changeability of the design in the event of context changes.

**TABLE 4-11:** Related Designs with the Cost Impact of Adding Change Mechanisms

Design Number	VWFO	Maneuver Level	Constellation Option	Lifecycle Cost (\$M)	Diff from Baseline (\$M)	Percent Diff
<b>Pareto Trace</b>						
<b>3435</b>	<b>-0.0057</b>	<b>4</b>	<b>1</b>	<b>14.29</b>		
3439	-0.0040	5	1	14.66	0.38	0.03
3443	0.0020	6	1	15.58	1.29	0.09
11211	-0.0027	4	2	17.11	2.82	0.20
11215	-0.0016	5	2	17.57	3.28	0.23
11219	-0.0007	6	2	18.65	4.36	0.31
18987	-0.0032	4	3	17.48	3.19	0.22
18991	-0.0020	5	3	17.99	3.70	0.26
18995	-0.0001	6	3	19.17	4.88	0.34
<b>Filtered Outdegree</b>						
1141	0.0021	4	1	13.27		
1145	0.0102	5	1	13.65	0.38	0.03
1149	0.0027	6	1	14.41	1.14	0.09
8917	0.0081	4	2	17.96	4.69	0.35
<b>8921</b>	<b>0.0147</b>	<b>5</b>	<b>2</b>	<b>18.51</b>	<b>5.25</b>	<b>0.40</b>
8925	0.0107	6	2	19.60	6.33	0.48
16693	0.0060	4	3	18.55	5.29	0.40
16697	0.0141	5	3	19.20	5.94	0.45
16701	0.0089	6	3	20.49	7.22	0.54
<b>Value Weighted Filtered Outdegree</b>						
1105	0.0058	4	1	31.19		
<b>1109</b>	<b>0.0183</b>	<b>5</b>	<b>1</b>	<b>31.84</b>	<b>0.65</b>	<b>0.02</b>
1113	0.0085	6	1	33.14	1.95	0.06
8881	0.0089	4	2	35.61	4.42	0.14
8885	0.0168	5	2	36.34	5.15	0.17
8889	0.0123	6	2	37.79	6.60	0.21
16657	0.0073	4	3	36.18	4.99	0.16
16661	0.0176	5	3	36.95	5.76	0.18
16665	0.0104	6	3	38.49	7.30	0.23



**FIGURE 4-16:** Spider Plot with Selected High Value Weighted Filtered Out-degree Designs

Notice that 1109, which was identified as having the highest VWFO, does not have the maximum level of the path enabling design variables, it is in fact in the middle. This suggests there is a threshold above which adding changeable options gives diminishing returns. A design may not utilize all the change mechanisms it carries to respond to the era context changes. This could be considered ‘extra’ flexibility, and because it is not valuable to the stakeholder in this particular era, the design is considered less flexible than other designs that have the ‘just right’ amount of flexibility. However, in a different era, during which the designs may encounter different context changes, the designs that previously had too much changeability may now be the just right designs. This case application analyzed a single era; studying additional eras are necessary to understand the complete interaction of design space and epoch context changes. In addition, the case application used a single design rule, ‘redesign’, which would not be applied once the satellite entered operations. This results in designs that have change mechanisms which are meant to be

deployed in operations are undervalued in the case application. This is discussed further in the future work suggestions in Chapter 6.

## 4.9 Summary and Limitations of VWFO

The SRS case application revealed several interesting aspects of VWFO. Several designs had high VWFO, and one way for a designer to determine which design to analyze further is to use Figure 4-12, which indicates designs with high VWFO and high origin design utility. By having positive VWFO and high starting utility, these designs have more transitions to other high utility designs.

There are several limitations to VWFO. The first is that the results obtained from any study of this nature will depend on the transition rules chosen. If the designers do not specify transition rules that are useful during context changes, no designs will be identified. In addition, it is also dependent on the order of epochs in the era, which determines which designs are valid for transition. Essentially, if a design is invalid in either epoch, it appears as invalid in both. Another problem with the metric occurs when the VWFO of a design is zero. The designer does not know, without further analysis, if that design has zero VWFO because it is an invalid design in one of the epochs, or because the net utility change is zero. The metric is also dependant on the tradespace sampling strategy used by the designer. If the design space has many designs in one area of the design space, which causes the FOD of those designs to increase, it is likely that the VWFO of the design will increase disproportionately as well.

Value Weighted Filtered Outdegree is able to identify valuably flexible SRS designs. The designs identified tend to have some level of change mechanism, but not the maximum. However the designer needs to be aware that results may be biased, i.e. the lower cost designs may appear to have less expensive transitions if the transition cost is assumed to be a percentage of the initial system cost.

**TABLE 4-12:** Design Vector for SRS of Interest

Design Number	Orbit Altitude (km)	Walker ID <sup>a</sup>	Antenna Area (m <sup>2</sup> )	Bandwidth (GHz)	Peak Power (kW)	Comm Arch. <sup>b</sup>	Total Utility
1089	800	4	40	0.5	1.5	0	0.6205
1101	800	4	40	0.5	10	0	0.7156
1105	800	4	40	0.5	20	0	0.7437
1109	800	4	40	0.5	20	0	0.7437
1113	800	4	40	0.5	20	0	0.7437
1141	800	4	40	1.0	20	0	0.7703
1145	800	4	40	1.0	20	0	0.7703
1149	800	4	40	1.0	20	0	0.7703
3435	1500	3	40	2.0	10	1	0.7535
3439	1500	3	40	2.0	10	1	0.7535
3443	1500	3	40	2.0	10	1	0.7535
3447	1500	3	40	2.0	20	1	0.7580
3555	1500	3	100	2.0	20	1	0.8152
6027	1200	3	40	2.0	10	1	0.7558
6039	1200	3	40	2.0	20	1	0.7642
6147	1200	3	100	2.0	20	1	0.8129
8881	800	4	40	0.5	20	0	0.7437
8885	800	4	40	0.5	20	0	0.7437
8889	800	4	40	0.5	20	0	0.7437
8917	800	4	40	1.0	20	0	0.7703
8921	800	4	40	1.0	20	0	0.7703
8925	800	4	40	1.0	20	0	0.7703
11211	1500	3	40	2.0	10	1	0.7535
11215	1500	3	40	2.0	10	1	0.7535
11219	1500	3	40	2.0	10	1	0.7535
16657	800	4	40	0.5	20	0	0.7437
16661	800	4	40	0.5	20	0	0.7437
16665	800	4	40	0.5	20	0	0.7437
16693	800	4	40	1.0	20	0	0.7703
16697	800	4	40	1.0	20	0	0.7703
16701	800	4	40	1.0	20	0	0.7703
18987	1500	3	40	2.0	10	1	0.7535
18991	1500	3	40	2.0	10	1	0.7535
18995	1500	3	40	2.0	10	1	0.7535

<sup>a</sup> Walker ID is a lookup table of characteristics for a Walker constellation, with different possibilities for number of satellites and the phase(plane orientation).

<sup>b</sup> Comm Architecture 0 = direct downlink, 1 = able to use relay backbone.

