Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis

Adam M. Ross
adamross@mit.edu
(617) 253-7061

Donna H. Rhodes
rhodes@mit.edu
(617) 324-0473

Massachusetts Institute of Technology
77 Massachusetts Avenue NE20-388
Cambridge, MA 02139

Abstract. A traditional approach to system design is to optimize the system with regard to a set of system objectives, as defined in a given context. This approach falls short when designing systems that are capable of delivering sustained value to stakeholders in the face of a rapidly changing world. In order to achieve this value robustness, systems should be designed using natural value-centric time scales, as defined by their contexts, for conceptualizing system timelines. *Epoch-Era Analysis* is an approach that provides for visualization and a structured way to think about the temporal system value environment. This paper discusses Epoch-Era Analysis as central to a tradespace exploration process for system design comparison and selection, invoking passive or active value robustness design strategies. The analysis can also serve as a socio-technical bridge, integrating the tradespace exploration activities of architects and engineers, which may be traditionally independent efforts in contemporary engineering programs.

Introduction

A common approach in system design is to optimize the system according to a set of objectives, where finding the "best" system is the goal. The technocratic optimization approach works well and will provide the "best" answer only if the assumptions of the analysis continue to hold true in the real world. Unfortunately, it is often true that such is not the case, as systems exist in a changing world and often come to operate in a context for which they were not originally intended when the system was initially conceived. A fundamental assumption in the optimization approach is the assumption of correct objective functions. If the objective functions change with time, no guarantee exists that the system will remain "best." In fact, recent research suggests that such overly optimized designs are quite fragile in the face of changing objectives or contexts (Carlson and Doyle 2000). Fricke and Schultz (2005) describe important principles for enabling systems to change throughout their lifecycle along with some enabling practices citing "dynamic marketplace," "technological evolution," and "variety of environments" as examples of new operating contexts and objectives for systems. The paper outlines a good set of heuristics for thinking about designing for changeability, but stops short of developing deeper analytic methods.

Related to the assumption of static objectives is that of static context. *Design for value robustness* is an approach used to search for design options that will continue to perform well in the face of changing operational environments and a dynamic context (Ross 2006). *Value*, in this sense, is a moving measure of success as defined individually by system stakeholders. “Best” designs create the most value in the minds of stakeholders over time. The system context, in the broader sense, includes not only its operational environment, but also stakeholder sets, the expectations of the system, the resources available to the system, the competition for the system, and any other exogenous factors that affect the perceived value of the system. Beyond consideration of the technical performance environment, many of these factors are not typically considered during an optimization exercise, though they may significantly affect the perceived value of the system. An example of such a dangerous omission in the dynamic context is a technical optimization of a satellite-based communication system, ignoring terrestrial-based competition, such as the fate suffered by the Iridium and Globalstar systems.

Successfully realizing the *design for value robustness* approach depends upon moving beyond traditional systems engineering and optimization practices, and invoking an enriched understanding of the temporal view of systems. This includes the incorporation of a dynamic perspective into the engineering practice, accommodating for differences in decisional and experienced value in decision making, and ensuring that tradespace exploration adequately considers the system’s external context. This approach can be realized through the use of natural value-centric time scales for conceptualizing system timelines. Natural value-centric time scales, mean temporal periods that are defined by their contexts, somewhat analogous to how historical or geologic epochs are defined by the cultural or environmental shifts within the era.

Whereas traditional systems engineering involves early phase tradespace exploration followed by sequential development, operation, and maintenance of a system, many modern systems demand an advanced systems engineering practice. In this case, tradespace exploration becomes a periodic activity throughout the system lifespan, often much longer than in previous systems, and is driven by the need to adapt the system for rapidly changing technologies, responsiveness paradigms, and participation in a system of systems endeavor. Three important considerations that inform the conceptualization of system timelines and an associated dynamic tradespace exploration approach include: (1) a static versus dynamic perspective; (2) decisional versus experienced value; and (3) parameterization of the tradespace.

Static versus Dynamic System Perspective

To understand the temporality of systems, one needs to distinguish between the static and dynamic system perspectives. The classical design approach is rooted in a static perspective, where the system is assumed to exist within a fixed context with well-defined goals and requirements. Given the recognition that modern complex systems will inevitably violate the static perspective assumptions, a dynamic systems perspective becomes more appropriate.

To first order, time can be incorporated into dynamic perspectives by considering the world to be similar to a movie composed of a series of static frames run in quick succession. The snapshot interpretation allows for a simple extension of static analysis techniques and perspectives. Figure 1 shows the relationship between the static and dynamic perspectives.

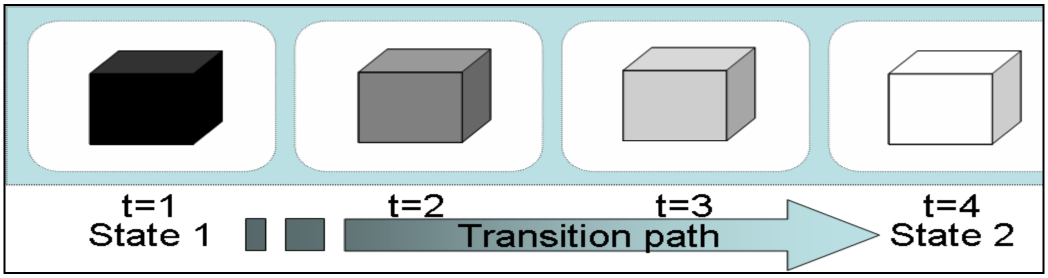

Figure 1. Time as a series of static snapshots.

Continuity of states across the boundary of the frames ensures coherent stringing of the static snapshots. In the limit each frame's duration approaches zero, the analysis becomes continuously dynamic. Wrapping time into the perspective of system development depicts the forces of value over time, which is an important consideration in trades and decision making. At any point in time, value is shaped by the many unique factors that define the system context in that period.

Decisional versus Experienced Value in Decision Making

Decision making is often seen from a static perspective, but in fact individuals are free to change their mind, and rightly so, given new information in a dynamic world. Therefore, it is important for designers to understand how decisional value changes over time, and to incorporate this knowledge into the design process. When confronted with a decision, a decision maker perceives his options through a "decisional" value lens, that is, from an envisioned experience rather than an actual one. Once the decision maker interacts with the system in use, the decision maker perceives his choice through an "experienced" value lens. These two lenses are not the same. In many development projects, the designers may use prototypes, mock-ups, and experiments to try to emulate the real experience, which is helpful. However, only use of the system in an operational context will result in actual "experienced" value.

Figure 2 below depicts the relationship between *decisional* and *experienced* value in relation to system design decisions in a dynamic context. *Experienced* value corresponds to the original Bentham meaning of utility (Bentham 1987) as relating to pleasure and pain, as opposed to more modern interpretations as a decision metric, which corresponds to *decisional* value. *Remembered* value is the recall of *experienced* value and typically informs the *decisional* value for a decision maker. Kahneman (2000) describes experiments and discusses motivations for these types of values, which he calls "utilities.".

A simple example of these two perspectives is a dilemma that confronts restaurant patrons on a daily basis. The patron reading the menu selects from the options using *decisional* value, which is an expectation of the experience with the ultimate choice. Upon eating the selected meal, the patron might realize that 'his eyes were too big' or find that the meal was not prepared to his taste, and as a result his *experienced* value was less than anticipated. If the patron learns from the experience, the *experienced* value will inform his *decisional* value and the next time the decision may be different, perhaps ordering a smaller portion size or an alternate preparation of the selected meal. In this example, the decision maker has the opportunity to learn and adjust future decisions. However, it should be recognized that there are systems where learning cannot occur, and in this case convergence between these two lenses cannot be expected.

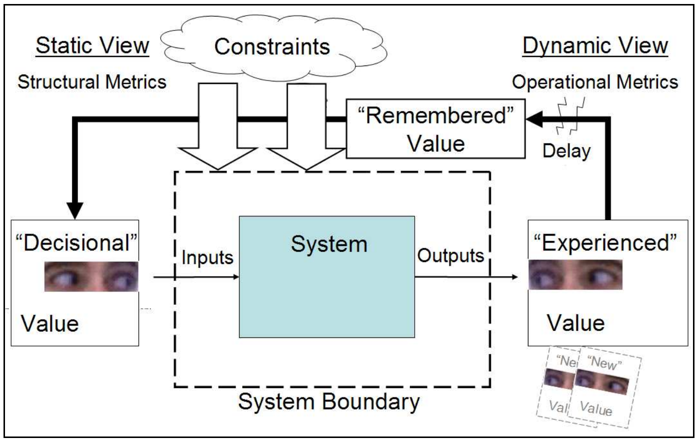

Figure 2. Dynamic value lenses: Decision versus Experienced Value.

One point to consider when making decisions about systems is to be aware of the nature of the metrics being used for the basis of the decisions. Structural metrics relate to the physical, or static, properties of the system. Operational metrics relate to the functional, or dynamic, properties of the system (Giachetti et al. 2003). Since these metrics may be assessed at different ends of the system choice timeline, two different value lenses may inadvertently be applied, resulting in inconsistent perspectives. Decisional and experienced values are fundamental concepts for tradespace exploration and for understanding how value perceptions shift over time.

Tradespace Parameterization of Systems

Quantification is a necessary step in order to develop a concrete specification for understanding system changeability across the system lifespan. A reasonable approach to comparing a large number of systems simultaneously is through a tradespace (Ross and Hastings 2005). Figure 3 depicts the elements that go into tradespace development. First, the designer has a dialogue with key stakeholders to identify system value and system concepts. Next, the value is parameterized as attributes of the system (aggregated in this example as utility). The concepts are parameterized as design variables (aggregated in terms of a broadly defined "cost"). The design variable set represents the physical degrees of freedom for the system, including only those facets of the design over which the designer has some control. Iterative analysis is performed, resulting in a tradespace plot of utility versus lifecycle cost, with each point in the graph representing a specific design alternative.

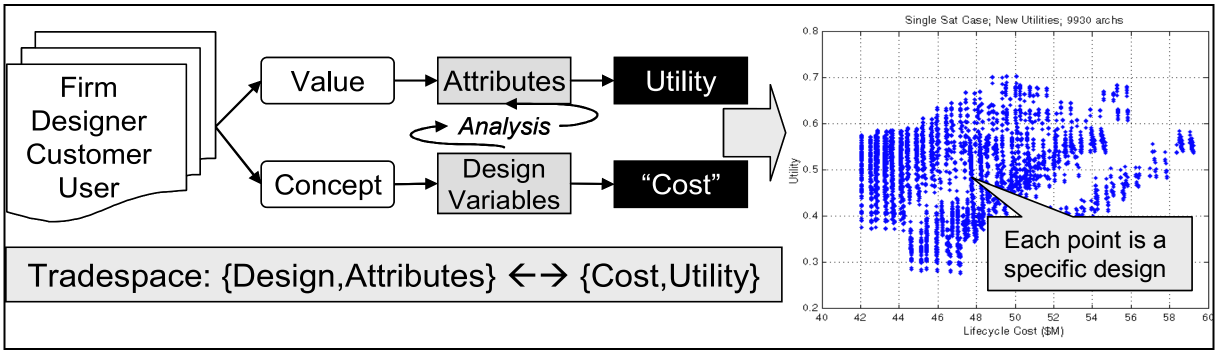

Figure 3. Tradespace defined as shorthand representations of designer controlled technical parameters and stakeholder perceived value parameters evaluated in terms of utility and cost.

Traditional systems engineering employs tradespace analysis in the early decisional phases in the system lifecycle, however these are typically oriented toward a static context and are rarely revisited in the lifecycle process. A dynamic tradespace exploration approach can help to realize value robust systems. This approach considers the system across contexts, or *epochs*, that are anticipated. Shifts in value expectations can be applied and the differential effects on design alternatives can be assessed. By adding consideration for how designs can move in a tradespace, tradespace exploration changes from being a 'front-end activity' in the sequential development process to a continuous mechanism for assessing system transition paths as contexts change, and for determining how the system can be transformed to address value shifts as it enters a new epoch.

Epoch-Era Analysis

Conceptualizing System Timelines

The *system lifecycle* is the core construct that designers use to characterize the phases of a system during its lifespan, from initial concept to end of life. A number of engineering standards and formal processes have been published to provide guidance on the system lifecycle (ISO 2002, INCOSE 2007). System lifecycle processes are beneficial to the designer for organizing the various activities required to design, develop, and operate a system. The system lifecycle is comprised of phases that have defined end points, but these are typically based on the resources available to complete a set of phase activities. Thus, the system lifecycle view is an imposed artificiality for managing system activities, and does not explicitly consider system context as a dynamic variable. Further, traditional systems engineering takes a limited view of systems context. Rhodes and Hastings (2004) describe the importance of an *engineering systems perspective* applied to systems engineering practice to provide the enriched view of context that is necessary for large scale complex systems.

The traditional conceptualization of system timelines does not consider impacts of a diverse set of context changes. Further, the system lifecycle does not enable understanding of system value delivery across its lifespan. This does not imply that the system lifecycle construct is non-useful; on the contrary it is essential for effective control and management of the system elements during its life. However, the designer must have other fundamental constructs for considering the temporal view to design highly complex systems that will be value robust to change. The static versus dynamic perspective and the 'decisional' versus 'experienced' value lens are constructs that inform the understanding of the temporal view. The parameterization of tradespaces provides an analytic basis to trade design alternatives in context of a temporal view.

Epoch-Era Analysis is an approach for conceptualizing system timelines using natural value centric timescales, wherein the context and expectations define the timescales. The full lifespan of system is referred to as the *System Era*, which can be decomposed into *Epochs.* An Epoch is a period of time for which the system has fixed context and fixed value expectations. Each epoch is characterized by static constraints, available design concepts, available technology, and articulated attributes. As changes trigger the start of a new epoch, the system may need to transform in order to sustain value, or else it may fail to meet expectations as defined for this new epoch, as illustrated in Figure 4 below.

Figure 4 illustrates the temporal evolution of a system as needs and contexts change. A system exists in Context 1 in Epoch 1 and has performance exceeding expectations. Expectations are represented by a band capturing the range from minimally acceptable to the highest of expectations. In Epoch 2, the context changes to Context 2 and the system when entering this context finds its performance is degraded. Yet, expectations are still met with the same system, so the system is relatively *robust* to the change in context. A change in expectation is shown in Epoch 3, with the context remaining the same as the second epoch; now the still unchanged system exhibits *value robustness* since it maintains value delivery in spite of changes in expectations. In Epoch 4, the system shows *versatility* by continuing to satisfy expectations despite the introduction of a new metric of need. Notice that even though the system no longer exceeds all expectations, it still does exceed the minimally acceptable level and thus is still successful. Finally, in Epoch 5, a change in context and a boost in expectations are too much for the system as-is; in this case the system must change in order to remain successful. If the system is capable of changing at acceptable cost, it is deemed *flexible* or *adaptable*, depending on the type of change desired (McManus et al., 2007).

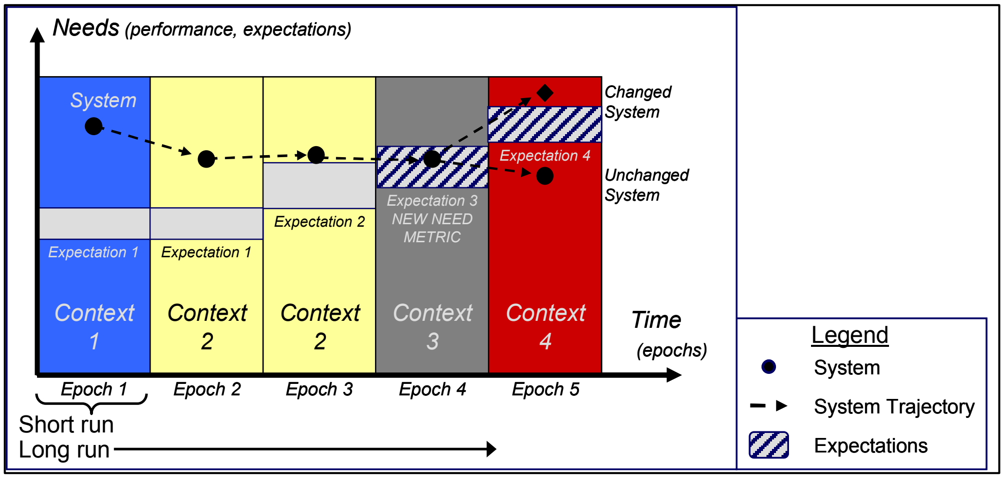

Figure 4. System Needs versus Expectations across Epochs of the System Era

Short Run and Long Run Analysis

In Economics, the difference between the short run and the long run, according to (Pindyck and Rubinfeld 2001) is that the short run is the "period of time in which quantities of one or more production factors cannot be changed," while the long run is the "amount of time needed to make all production inputs variable." The key distinction is that in the short run, some aspects of the analysis are fixed, while in the long run, everything is variable. Applying this concept to the analysis of systems over time results in the analogical comparison as shown in Table 1.

Table 1. Fixed factors for analysis in Economics and Engineering for Short and Long runs.

	Short run		**Long run**	
	Fixed factors	**Variable factors**	**Fixed factors**	**Variable factors**
Economic Analysis	Input factors such as labor, capital equipment, regulations, knowledge	Quantity demanded, quantity supplied, dollars and time spent	none	all
Engineering Analysis	Objectives, constraints, stakeholder set, technology, concepts	Design chosen, dollars and time spent, perceived value	none	all

Bringing in the concepts of context and time, and borrowing from economics, a system change scenario can be defined in terms of epochs. As noted above, the *Epoch* is a time period that bounds the change scenario, during which utility functions, constraints, design concepts, available technologies, and articulated attributes are defined. The purpose of the epoch is similar to short run analysis in Economics: to parse a complex problem into a series of simpler ones.

Similar to the economic short and long run analyses, many system aspects and constraints are "fixed" in the short run (epoch), but variable in the long run (era). For each epoch i, the duration of the epoch, as well as the beginning state, $S_{i,b}$, and ending state, $S_{i,e}$, must be defined. (If no change occurs, the two design states are the same.) Continuity of states means that the ending state of epoch i is equal to the beginning state of epoch $i+1$. Figure 5 shows a qualitative picture of a string of epochs, which together form the system era.

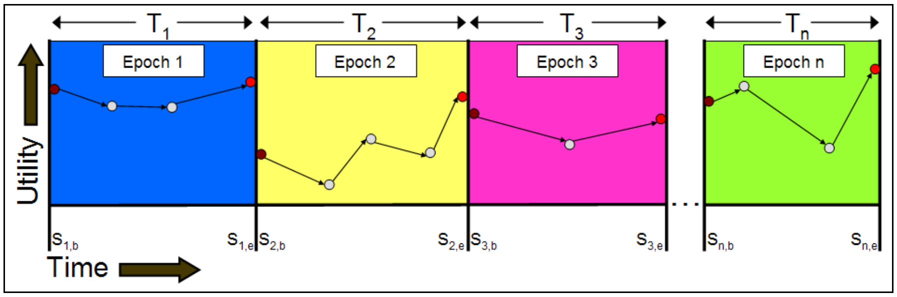

Figure 5. Example of Epoch string to form System Era.

An example system era for a satellite system with serviceability is shown in Figure 6. Epoch 1 begins with the system beginning of life (BOL), and as the system operates within the static context of this epoch, value degradation may occur and in this example, a major failure then occurs. There is service to restore the satellite to the prior level of value delivery in its continued operation. Then, a new context begins in Epoch 2, perhaps involving the introduction and expectations of a new technology, resulting in the same system being perceived by the stakeholders as having significantly decreased value. After a service to upgrade it with the new technology, possibly including a service time outage, the system again reaches a high level of perceived value. The system continues its operation in Epoch 2 until context changes result in another epoch. This continues until end of life (EOL).

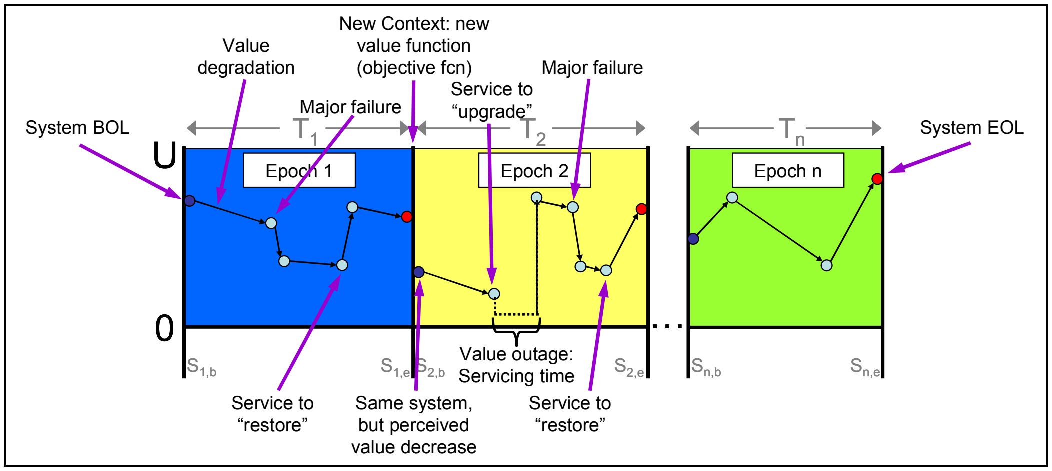

Figure 6. Example Era for satellite with serviceability.

During each epoch in the system era, path analysis can be conducted, utilizing an *accessibility matrix* generated in the tradespace exploration process, which denotes which designs in the tradespace are reachable from another design in the tradespace by application of a *transition rule* (Ross and Hastings 2006). A goal design can be specified, and paths from a baseline design to the goal can be derived using any specified transition strategy. Figure 7 shows an example for a single epoch, where the strategy is to find the minimum cost path from a start state (design 81) to an ending high utility state (Utility=0.97).

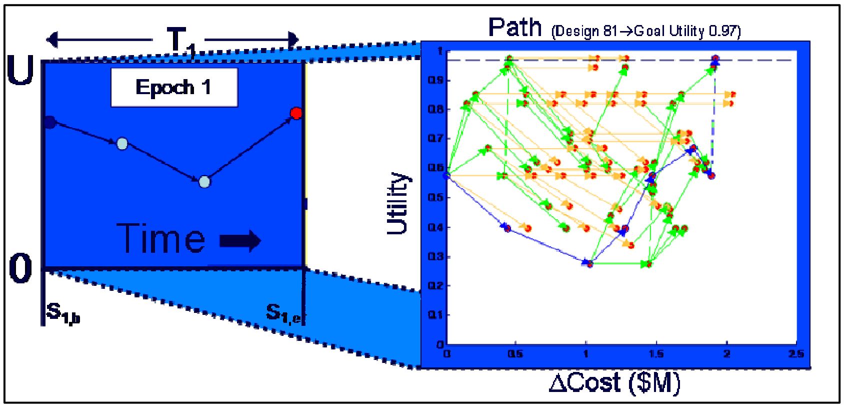

Figure 7. Example notional one Epoch path analysis.

The beginning design state $S_{1,b}$ is specified, along with the preferences, the design concepts, constraints, available technology, and epoch duration. The accessibility matrix is calculated using the available transition rules defined for each point design. Allowable paths are investigated and the "best" path is determined according to the specified strategy (e.g. minimum cost or maximum utility, etc.) in order to reach the goal state $S_{1,e}$.

Now, suppose a new expectation for the system arises, necessitating the addition of a second epoch. The beginning state for the second epoch is the same as the ending state of the previous epoch. Even though the system is physically and functionally the same, the system could be perceived to have lower value due to perception shift in the new epoch. Figure 8 shows the adjustment of utility in the same system under the new context. The designer now tries to find another path in the new epoch in order to increase the value of the system. Again, path analysis can be readily applied to the tradespace.

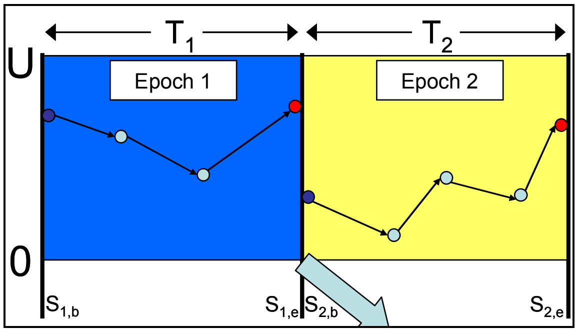

Figure 8. Two Epoch path analysis with utility discontinuity due to new expectations.

It should be noted that the depicted utility drop across Epochs is only as a typical example of what may happen when a changed set of preferences is levied on a system. In some cases, the new preference (expectations) may already be met by the system (a case of latent value in the system) and the utility may NOT drop across the Epoch boundary; it may even increase. For example, "*the system already had the ability to give me X, even though I didn't know I needed it, but now I do.*"

Considerations for Epoch-Era Analysis

There are a number of important considerations for performing Epoch-Era Analysis, including:

1. The Epoch timelines can be assessed at any point during the system lifecycle, not only during early conceptual design. As previously noted, this aspect enriches the traditional systems engineering practice of performing static tradespace analysis in the front-end concept phase. Tradespace exploration can be performed when a new epoch is recognized or anticipated. For analysis purposes, epochs can be known in advance, or in the moment, and can be deterministic, or probabilistic. As such, the mathematical treatment of the paths, costs, utilities, and times must appropriately match the uncertainty level of the data.

2. Value (expressed as utility) is determined during each epoch since perception is context dependent and varies with time. Strictly speaking, within an individual epoch the available technology is fixed, but can vary across the boundary from epoch to epoch, however. Given the dynamics of reality, the anticipation of new technologies may result in new (and perhaps higher) expectations on a system; however, this increase in expectations may not coincide with the availability of the actual technology. This results in a decrease in perceived utility during the gap between expectation and experience on that technology. Once that technology is available, a new epoch begins, and the utility may jump as soon as it is incorporated into the system. It is important to note that utility is only well defined within a given epoch; the absolute numbers cannot be compared across epochs. It may be the case in absolute terms that a user is much more pleased tomorrow, given a new technology, than today. But ignorant of tomorrow's technology, that same user "could be perfectly happy today."

3. Selection of the system Epoch end state is dependent on the strategy for the Epoch. No absolute correct or "best" design exists without subjectively specifying the "best" strategy. Strategies can include seeking maximum utility, minimum cost, minimum time, minimum risk, or any combination, among others.

4. Strategies themselves can be predictive, adaptive, or static, meaning an analyst can use them to predict "best" paths for a system given present knowledge of the future, to adapt given new information about current and future epochs, or to statically drive a particular agenda for a fixed set of objectives and technology in a changing world. The system analyst can use the epoch analysis while the system is in operation, continuously updating probabilities and value data to determine the "best" path to other designs, as well as the "best" goal design to pursue in each epoch.

5. The epoch is a mechanism for stringing together short runs (Epochs) into the long run (System Era), and as such, it may make sense to sometimes "take the longer view" rather than only seek the best solution for a given epoch. It is possible, due to the path-dependence of design transitions, that some future "best" state may only be accessible by accepting the "bad" designs in the short run. Such insight can be gained using this analysis approach.

6. The modularity of the Epoch-Era Analysis approach enhances overall tradespace exploration. Since a “full” tradespace is generated in each Epoch, and the costs and utilities plotted are independent of design alternatives, the epoch tradespace network is not dependent on the starting, or baseline design chosen for that epoch. As such, system era timeline analysis is modular by design. Various epochs can be specified in advance and strung together in various orders, since order is not always known. Paths within and across these timelines can be calculated according to arbitrary strategies. In this way, system era timeline analysis is reusable, with the analyst able to update calculations and strategies as new information becomes available.

7. At least two distinct approaches for achieving value robustness can be identified using the Epoch-Era Analysis, as shown in Figure 9 below. One approach is to minimize the need for changeability, wherein the designer seeks designs with no need to transition, which remain high value in spite of epoch changes. This approach identifies designs most “passive value robust,” where the “clever design” continues to deliver value as context and expectations changes occur. The second approach is to maximize the need for changeability, wherein the designer seeks designs with the ability to transition to other designs, which allow the system to move to high value perception across epoch changes. This latter approach identifies “active value robust” designs.

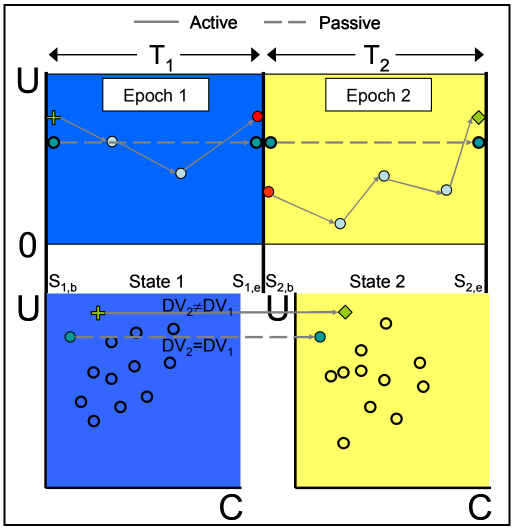

Figure 9. Active versus passive value robustness across two Epochs.

Epoch-Era Analysis Method as Boundary Spanning Activity

The practice of conducting tradespace studies in the design of a system often occurs in two non-intersecting activities. The first instance of tradespace studying occurs at the conceptual level through the use of ‘storyboarding’ to define high-level scenarios. In this case, the activity is typically performed by system architects for the purpose of defining system concepts and communicating with stakeholders in an effort to learn more about their needs, expectations, and preferences. The outcome of this type of tradespace study is usually a set of “cartoon-like” graphics, and narrative operational concepts or scenario descriptions. Kazmen et al. (2000) describe the importance of scenario-based analysis in software architecting, advocating the approach of linking high level scenarios with analytic models, calling for a discipline of "architectural engineering" wherein design decisions are motivated by scenarios and supported by design analysis.

The second instance of tradespace studying uses deep analytic methods, usually modeling and simulation based. In this case, the activity is performed by technical engineering specialists and analysts, and the outcome is in the form of a model and/or a highly technical report. A shortfall of current approaches is that the activities are for the most part independent, and focus on two distinct aspects of the design. Decoupled, the separate approaches miss opportunities for informing architectural and design choices of mutual benefit. Epoch-Era Analysis as a tradespace exploration practice is boundary spanning, with an integrated process:

1. First, possible scenarios and architectural trajectories are defined by the system architecting team for the epoch(s), and shared with the analysts.
2. Working together, the architect(s) and analysts detail various characteristics and variables to be used in the deeper analysis, as shown in Table 2 below.
3. Next, the analysts develop the detailed models in context of the Epoch-Era framework using the defined set characteristics and variables.
4. Software is then used to produce graphic visualizations of large numbers of designs that are readily understood and interpretable.
5. The resulting information is at the level that it can be used to enable effective communication between system architect, system stakeholders, and analysts. Thus, the outcome of the activity can now be used for an effective dialogue among all of the stakeholders engaged in exploring the design options.

A strength of Epoch-Era Analysis is that it is suited to the interests and information preferences of both the architect and the analyst. It accommodates and brings together both the architect's architecture trajectory viewpoint and the analyst's point design viewpoint. As such, it provides a unifying method so that decisions can be mutually explored and determined. As described in the next section of the paper, this is aided by the shared set of constructs and descriptive elements used to capture information for the analysis.

Capturing Information for Epoch-Era Analysis

Developing the information for a given epoch, the set of information to be captured, includes:

- Epoch Identifier, describing the duration and goals of the epoch
- Constraints, characterizing the imposed fixed boundaries for consideration of the epoch including resource, political, market, operational, physical and other constraints
- Constants, describing design variables which do not change within the epoch, including those set up the designer or those beyond the control of the designer
- Preference Space, detailing the decision makers, and for each, their attributes, attribute priorities, utility functions, and changeability acceptability thresholds
- Design Space, details the design variable set, baseline design, path enabling parameters, transition rules, and cost functions
- Model Space, describes any visual or descriptive scenario that informs the analysis, and the analytic model(s) used in the analysis.

Table 2 describes the full set of information that is needed to characterize an epoch for analysis.

Table 2. Example Epoch-Era Analysis Descriptive Element Template.

Analysis for X-TOS System Era – Epoch n	
Epoch Identifier	**Description**
Epoch Name	The descriptive name for the epoch, for example: ***X-TOS Initial Operating Scenario***
Epoch Duration	Finite duration for the epoch, for example: five years, or until system context change
Epoch Goal	Overall goal for epoch, for example: **Find Maximum Utility Design At** $S_{i,e}$
Constraints	**Description**
Resource	All of the resource related constraints including time, financial, manpower, and others, for example: *Must spend less than $100M over 5 years*
Political	The political related constraints which may be by formal policy or implicit, for example: *Must not use foreign launch vehicle*
Market	Market constraints including limitations imposed and windows of opportunity
Physical	Physical system constraints including limits by physical laws, spatial limits, etc.
Operational	The operational constraints in regard to system performance and other operating considerations, for example: *Must provide less than 5 Gbps downlink data rate*
Other	Any other constraints not enumerated in the previous categories.
Constants	**Description**
Constant Variable Set {CON}	The set of design variables that is constant within this epoch.
Controllable	The constants which are controllable by the designer.
Uncontrollable	The constants which are beyond the control of the designer.
Preference Space	**Description**
Decision Maker set, {DM}	The set of decision makers for the epoch, for example: *system user*
Number of DM {DM}	The number of decision makers for the epoch, for example 1
Attribute set, $\{X^{M}{}_{i}\}$:	Attribute set for the epoch, defined for each decision maker i. For example: {Data Lifespan, Latitude Diversity, Equator Time, Latency, Sample Altitude}
Attribute Priorities, $\{k^{M}{}_{i}\}$:	The priorities on a scale of 0 to 1, defined for each decision maker, for example: [0.3,0.125,0.175,0.1,0.425]
Single attribute utility curves, $U_j(X_j)$:	Single attribute utility curves for the epoch for each decision maker.
Multi-attribute utility curve, $f(U(k_j,U_j))$: *MAUF*	Multi-attribute utility curves for the epoch for each decision maker.
Changeability Cost threshold, $\hat{C}$:	The highest level change cost that a decision maker is willing to accept, for example $50M
Changeability Time threshold, $t\hat{}$:	Changeability acceptable time threshold of a decision maker; this varies if making decision for short term (this epoch only) or longer term (multi-epochs).
Design Space	**Description**
Design variable set, $\{DV^{N}\}$:	The set of design variables for the epoch, For example {Inclination, Apogee Altitude, Perigee Altitude, Communication Arch, Total DeltaV, Propulsion Type, Power Type, Antenna Gain}
Baseline design, DV_{base}:	Baseline design from the previous epoch; if this is first epoch then there is no baseline.
Path-enabling variable set, $\{IV^{P}\}$:	The variable set whose purpose is to enable new paths in the epoch tradespace, lowering transition cost or adding paths.
Transition rule set, $\{R^{K}\}$:	Rules for how to change design variable values, where change in one may result in change in another. For example, R1: Plane Change (burn on-board fuel to alter inclination), R2: Apogee Burn (burn on-board fuel to alter apogee), etc.
Cost function, F_c *(CON, DV, IV)*	The cost function for the design, based on the constants, design variables, and the path enabling variables, for example mass-based cost-estimating relationships.
Model	**Description**
Scenario	Visual and descriptive scenario, developed by Team ABC
Model(s) to be used, F_{XM}	The model(s) to be used, for example X-TOS code version 1.1 developed by Team XYZ.

Discussion

This paper introduced the concept of Epoch-Era Analysis as an approach for visualizing, communicating, and analyzing systems timelines through changing contexts. This area of research is important, as current analysis methods fall short of what is needed to address designing systems that will be capable of sustained value delivery in a dynamic world. Further, a gap exists between the concept tradespace exploration activity performed by system architects and that which is performed by analysts using detailed approaches and models. To address these concerns, the authors are pursuing four research directions as extensions of the work presented in this paper.

The first research direction involves deeper exploration of system context and decision analysis processes. Epoch-Era Analysis enables a more rigorous exploration from the temporal view, and there are a number of associated concepts and constructs to be further explored that will support this type of analysis. An area of interest is the decision making processes of individuals and small groups in making system concept, architectural, and design decisions. Related to this area of decision making is the need to examine the impacts of cognitive biases and impacts of different decision support environments.

The second research direction is evaluation of specific techniques that have emerged for time-based analysis, and their potential for augmenting Epoch-Era Analysis. Foundational concepts of the metasystems methodology presented by (Hall 1989) are of interest, wherein his vision was for a synthesis of systems methods to seek "value truths by matching the properties of wanted systems, and their parts, to perform harmoniously with their full environments over their entire lifecycles." *Scenario analysis* is one technique that has been used effectively as a technique by software engineering. Scenarios are "narratives of expected or anticipated use of a system from both development and end-user viewpoints" (Kazman, et al, 2000). Scenario analysis is typically used as part of an overall requirements engineering process for a single development effort, rather than for more conceptual and continuous tradespace exploration. Boardman (et al 2005) describe the use of the Systemigram™ Technique for visualizing a system of systems, including its usefulness as a tool for dialog. This technique is one example of formalization of tradespace exploration outcomes by architects.

The third area seeks to further develop and validate Epoch-Era Analysis on real world programs, including systems and systems of systems. An upcoming project funded by the US Government will conduct a large-scale Epoch-Era Analysis study of a complex space system architecture. It is anticipated that the study will result in further innovations, for example as a method for quantifying, visualizing, and communicating "ilities." McManus et al. (2007) describes some of the early thinking on this topic. Using Epoch-Era Analysis for quantifying "ilities" will likely provide leverage for designers to better understand how to adjust the path-enabling variable set, that is, to be able to actively design for "ilities" such as flexibility, versatility, agility, and survivability.

A fourth research direction focuses on creating educational innovations and thinking constructs to enable a shift in the engineering mindset toward the value robustness paradigm. This research includes performing case studies, conducting research with industry, developing teaching cases, and capturing new knowledge based on trial use of the method in the field. Additionally, it is important to understand where this type of analysis could be inserted into ongoing engineering education classes in academia and in the practitioner community. In particular, there is a need to assess education mechanisms for informing senior decision makers, as new methods and analysis approaches will not be effective unless practitioners at multiple levels are informed and educated on appropriate use.

Conclusion

The constructs of system *era* and *epoch* provide a mechanism for effective temporal tradespace exploration for engineering systems under the *value robustness* paradigm. Exploration of the tradespace across the entire system era and its anticipated (and actual emergent) epochs provides a mechanism for assessing system transition paths as contexts and expectations change. Epoch-Era Analysis serves, first of all, to provide qualitative constructs for visualization and communication of value over time in shifting epochs. Rigorous era analysis provides the quantitative construct for parsing the problem into simpler ones, wherein expected value calculations and other analysis can be performed. Together, these enable multi-level analysis for system options to sustain value delivery within and across changing epochs. Furthermore, the analysis provides a method to bring together the work of architects and analysts, the former having an architectural trajectory view and the latter having a discrete point design view. The use of Epoch-Era Analysis, as part of tradespace exploration, provides a means for the natural extension of static views to the dynamic view that is essential for designing systems for changeability.

The Epoch-Era Analytic approach will enable the designer to think in a more continuous and anticipatory manner in a world that demands engineering of highly complex systems. Of course, the designer is limited in predictive and cognitive capability, and therefore cannot predict all future scenarios. As such, the qualitative frame and quantitative approach selected will limit the valuation results. Nonetheless, Epoch-Era Analysis does significantly expand the tradespace exploration process. This analysis greatly enriches the temporal viewpoint of the systems designer, providing a natural value-centric scale for system timeline conceptualization, and allowing for the discovery and development of value robust systems in the contemporary environment. An additional benefit is that Epoch-Era Analysis can be implemented as a boundary spanning activity to integrate the conceptual level tradespace exploration of system architects and the deep analytic results produced by technical experts and modelers.

References

Boardman, J, Wilson, M., and Fairbairn, A., "Addressing the System of Systems Challenge", INCOSE International Symposium, Rochester, NY, July 2005

Bentham, J., "From an Introduction to the Principles of Morals and Legislation" in Utilitarianism and Other Essays. J.S. Mill and J. Bentham. New York, NY, Penguin Books: 65-112, 1987

Carlson, J. M. and Doyle, J., "Highly Optimized Tolerance: Robustness and Design in Complex Systems." Physical Review Letters **84**(11): 2529-2532, 2000

Fricke, E. and Schulz, A., "Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout Their Entire Lifecycle." Systems Engineering **8**(4): 342-359, 2005

Giachetti, R. E., et al., "Analysis of the structural measures of flexibility using a measurement theoretical framework." International Journal of Production Economics **86**: 47-62, 2003

Hall, A. D. III, *Metasystems Methodology*, Oxford, England: Pergamon Press, 1989

INCOSE, INCOSE Systems Engineering Handbook, v 3.1, INCOSE 2007

ISO, ISO/IEC 15288: 2002, Systems Engineering - Life Cycle Processes, International Standards Organization (ISO), 2002

Kahneman, D., "Experienced Utility and Objective Happiness: A Moment-Based Approach." in Choices, Values, and Frames. D. Kahneman. Cambridge, UK, Cambridge University Press**:**

673-708, 2000

Kazman, R, Carriere, S.J., Woods, S., Toward a Discipline of Scenario-based Architectural Engineering, Annals of Software Engineering, Vol. 9, 5-33, 2000

McManus, H.M., Richards, M.G., Ross, A.M., and Hastings, D.E., "A Framework for Incorporating "ilities" in Tradespace Studies," *AIAA Space 2007*, Long Beach, CA. September 2007

Pindyck, R. S. and Rubinfeld, D. L., Microeconomics. Saddle River, NJ, Prentice Hall, 2001

Rhodes, D. and Hastings, D., The Case for Evolving Systems Engineering as a Field within Engineering Systems, MIT Engineering Systems Symposium, 2004

Ross, A. M., "Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration", *Engineering System Division*. Cambridge, Massachusetts Institute of Technology**:** 361. 2006

Ross, A. M. and Hastings, D.E., "The Tradespace Exploration Paradigm". *INCOSE International Symposium*. Rochester, NY. June 2005

Ross, A. M. and Hastings, D.E., "Assessing Changeability in Aerospace Systems Architecting and Design Using Dynamic Multi-Attribute Tradespace Exploration". *AIAA Space 2006*. San Jose, CA. September 2006

Ross, A.M., Rhodes, D.H., and Hastings, D.E., "Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, and Robustness". *INCOSE International Symposium*. San Diego, CA. June 2007

Acknowledgement

Funding for this work was provided by the Systems Engineering Advancement Research Initiative (SEAri), a consortium of systems engineering leaders from industry, government and academia (`http://seari.mit.edu`).

Biography

Adam M. Ross. Dr. Adam M. Ross is a Research Scientist in the Engineering Systems Division (ESD) at MIT, and leads research in the MIT Systems Engineering Advancement Research Initiative (SEAri). His research focuses on managing unarticulated value, designing for changeability, and dynamic tradespace exploration for complex systems. Dr. Ross received his Ph.D. degree from MIT ESD, and was previously a postdoctoral researcher with ESD and a research assistant with the Lean Aerospace Initiative at MIT. Dr. Ross has published papers in the area of space systems design and changeability. He has work experience with government, industry, and academia including NASA, Smithsonian, Boeing, MIT, Harvard, and Florida State.

Donna H. Rhodes. Dr. Donna H. Rhodes is a Senior Lecturer and Principal Researcher in the MIT Engineering Systems Division, and research lead for the MIT Systems Engineering Advancement Research Initiative (SEAri). Her areas of specialization include technical and management theory, methods, and practices for architecting and design of complex systems, systems-of-systems, and enterprises. Prior to joining MIT, Dr. Rhodes had 20 years of experience in the aerospace, defense systems, systems integration, and commercial product industries. Dr. Rhodes is a Past President and Fellow of INCOSE, and is a recipient of the INCOSE Founders Award and several INCOSE Distinguished Service Awards. She has published numerous papers and research reports in the field of systems, and has co-authored industry and corporate engineering policies, standards, and technical reports.