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

Dr. Adam M. Ross and Dr. Donna H. Rhodes
MIT
Paper Session 11, Track 1: Process 2
Thursday, June 19, 2008

Presented to the INCOSE 2008 Symposium
Meeting Customer Needs

- Goal of design is to create value (profits, usefulness, voice of the customer, etc…)
- Requirements capture a mapping of needs to specifications to guide design
Deploying a “Valuable” System…

Contexts change…
Meeting Customer Needs (cont.)

• Goal of design is to create value (profits, usefulness, voice of the customer, etc…)

• People change their minds; the world changes…
• To continue to deliver value, systems must cope with context change…
What is Context?

• Context includes forces exogenous to system
  – Stakeholder expectations
  – Operating environment
  – Policy constraints
  – Available technologies
  – Competitive market
  – Etc…

• System success depends on system performance within a given context

• In order to ensure success, designers must consider context beyond traditional “operating environment” (classical robust design)
What is System Success?

Success is defined as meeting or exceeding decision maker expectations

\[ \Psi = \left[ X_{DM} \geq Y_{DM} \right] \]

\[ 0 \leq \Psi \leq 1 \]

System Success: Net “experience” must meet or exceed net “expectations”

Note: this is a simplistic version of a utility function (is the system attribute \( X \) above minimum acceptable level \( Y \)?)
What is System Success?

Success is defined across **multiple time periods**

<table>
<thead>
<tr>
<th>System success, $\Psi$, across time $t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>“experience”</td>
</tr>
<tr>
<td>$\Psi(t) = \left[ X_{DM}(t) \geq Y_{DM}(t) \right]$</td>
</tr>
<tr>
<td>$0 \leq \Psi(t) \leq 1$</td>
</tr>
</tbody>
</table>

$X_{DM}(t)$  | Decision maker system “experience” at time $t$ |
$Y_{DM}(t)$  | Decision maker system “expectation” at time $t$ |

Note: this is a simplistic version of a dynamic utility function (is the system attribute $X(t)$ above minimum acceptable level $Y(t)$?)

System Success: Net “experience” must meet or exceed net “expectations”
What is System Success?

Success is defined across multiple time periods and multiple perspectives.

System success, $\Psi$, across $N$ decision makers at time $t$

$$\Psi(t) = \sum_{i=1}^{N} \left[ X_{DMi}(t) \geq Y_{DMi}(t) \right]$$

$0 \leq \Psi(t) \leq N$

$X_{DMi}(t)$ Decision maker $i$ unaffected system “experience” at time $t$

$Y_{DMi}(t)$ Decision maker $i$ unaffected system “expectation” at time $t$

Note: this is a simplistic version of a dynamic utility function (is the system attribute $X(t)$ above minimum acceptable level $Y(t)$ for each decision maker?)

System Success: Net “experience” must meet or exceed net “expectations”
What is System Success?

Success is defined across **multiple time periods** and **multiple perspectives**

System success, $\Psi$, across $N$ decision makers at time $t$

\[
\Psi(t) = \sum_{i=1}^{N} \left[ X_{DMi}(t) + \varepsilon_C^{X_{DMi}}(t) \right] - \left[ Y_{DMi}(t) + \varepsilon_C^{Y_{DMi}}(t) \right]
\]

where

- $X_{DMi}(t)$: Decision maker $i$ unaffected system “experience” at time $t$
- $Y_{DMi}(t)$: Decision maker $i$ unaffected system “expectation” at time $t$
- $\varepsilon_C^{X_{DMi}}(t)$: Context effect on decision maker $i$ “experience” at time $t$
- $\varepsilon_C^{Y_{DMi}}(t)$: Context effect on decision maker $i$ “expectation” at time $t$

$0 \leq \Psi(t) \leq N$

System Success: Net “experience” must meet or exceed net “expectations”
Traditional Perspective

- Does my System...
  - have good requirements? (stable, achievable, verifiable, etc. across many use-cases, stakeholders, and environments)
  - meet the requirements?

- Does my System program have acceptable...
  - cost, schedule, risk, etc...?

**Changes are considered to be “bad”**

Inherently a “static” perspective, but methods bias us in this direction
If classical approaches bias us in the direction of a static perspective, how can we move closer to a dynamic reality?

View time as a movie reel (series of static boxes)

A string of static analyses can approximate dynamic analyses, in the limit
**“Decision”, “Experienced”, and “Remembered” Utility from (Kahneman and Tversky 2000)**

**Discussion of “structural” versus “operational” metrics in (Giachetti et al. 2003)**

seari.mit.edu © 2008 Massachusetts Institute of Technology
Tradespace Exploration

Many system designs can be compared through *tradespace exploration*:

1. Elicit “Value” with attributes and utility
2. Generate “Concepts” through design variables and cost models
3. Develop models/sims to assess designs in terms of cost and utility

Models and simulations determine attribute “performance” of many designs (1000s to 10000s or more)

**ATTRIBUTES:**
- Design decision metrics
  - Data Lifespan (yrs)
  - Equatorial Time (hrs/day)
  - Latency (hrs)
  - Latitude Diversity (deg)
  - Sample Altitude (km)

**DESIGN VARIABLES:**
- Design trade parameters
  - Orbital Parameters
    - Apogee Altitude (km)
    - Perigee Altitude (km)
    - Orbit Inclination (deg)
  - Spacecraft Parameters
    - Antenna Gain
    - Communication Architecture
    - Propulsion Type
    - Power Type
    - Total Delta V

---

*Example*

X-TOS
Small low-altitude science mission
Tradespace Networks

Transition rules are mechanisms to change one design into another. The more outgoing arcs, the more potential change mechanisms.

Tradespace designs = nodes
Applied transition rules = arcs
# Tradespace Networks

## Example: X-TOS Transition Rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
<th>Change agent origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: Plane Change</td>
<td>Increase/decrease inclination, decrease ΔV</td>
<td>Internal (Adaptable)</td>
</tr>
<tr>
<td>R2: Apogee Burn</td>
<td>Increase/decrease apogee, decrease ΔV</td>
<td>Internal (Adaptable)</td>
</tr>
<tr>
<td>R3: Perigee Burn</td>
<td>Increase/decrease perigee, decrease ΔV</td>
<td>Internal (Adaptable)</td>
</tr>
<tr>
<td>R4: Plane Tug</td>
<td>Increase/decrease inclination, requires “tugable”</td>
<td>External (Flexible)</td>
</tr>
<tr>
<td>R5: Apogee Tug</td>
<td>Increase/decrease apogee, requires “tugable”</td>
<td>External (Flexible)</td>
</tr>
<tr>
<td>R6: Perigee Tug</td>
<td>Increase/decrease perigee, requires “tugable”</td>
<td>External (Flexible)</td>
</tr>
<tr>
<td>R7: Space Refuel</td>
<td>Increase ΔV, requires “refuelable”</td>
<td>External (Flexible)</td>
</tr>
<tr>
<td>R8: Add Sat</td>
<td>Change all orbit, ΔV</td>
<td>External (Flexible)</td>
</tr>
</tbody>
</table>

Tradespace designs = nodes

Applied transition rules = arcs

Transition rules are mechanisms to change one design into another. The more outgoing arcs, the more potential change mechanisms.
Epoch-Era Analysis
The System Change Timeline

**Epoch**

*Time period with a fixed “context”*

**Fixed:** Constraints, design concepts, available technology, and expectations (attributes and utility function)

One Epoch: short run

Multiple Epochs (System Era): long run

**Legend:**

\( T_i \): Duration of Epoch \( i \)

\( S_{i,b}, S_{i,e} \): System State at beginning, end of Epoch \( i \)

Continuity of States: \( S_{i,e} = S_{i+1,b} \)

**Epoch Purpose**

Partition problem into series of short run problems
What Can Change: Short Run and Long Run Analysis

In economics, the *short run* is the decision-making period during which at least one input factor is considered fixed.

*Example*: lease on facility (commitment can vary 6 mos to 10 yrs or more)

The *long run* is the decision-making period considered when all inputs are variable.

**Implication**: in short run, optimize subject to fixed constraints; in long run, make strategic decisions to shift possibilities.

<table>
<thead>
<tr>
<th>Short run</th>
<th>Long run</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed factors</td>
<td>Variable factors</td>
</tr>
<tr>
<td>Economic Analysis</td>
<td></td>
</tr>
<tr>
<td>Input factors such as</td>
<td>Quantity demanded,</td>
</tr>
<tr>
<td>labor, capital</td>
<td>quantity supplied,</td>
</tr>
<tr>
<td>equipment, regulations,</td>
<td>dollars and time spent</td>
</tr>
<tr>
<td>knowledge</td>
<td></td>
</tr>
<tr>
<td>Engineering Analysis</td>
<td>Design chosen, dollars</td>
</tr>
<tr>
<td>Objectives, constraints,</td>
<td>and time spent,</td>
</tr>
<tr>
<td>stakeholder set,</td>
<td>perceived value</td>
</tr>
<tr>
<td>technology, concepts</td>
<td></td>
</tr>
</tbody>
</table>
Capturing “Context” Effects on Systems over Time

- Plot performance/expectations vs. time
- Context, system, expectations change in epochs

**Epoch:**
Time period with a fixed context; characterized by static constraints, design concepts, available technologies, and articulated attributes (Ross 2006)

**Legend**
- ● System
- ● System Trajectory
- ← Expectations

Discretization of change timeline into short run and long run enables analysis
Using Epochs to Represent Context and Expectations

Two aspects to an Epoch:

1. Needs (expectations)
2. Context (constraints including resources, technology, etc.)

Example system: Serviceable satellite

System timeline with “serviceability”-enabled paths allow continued value delivery
Epoch-Era Analysis: Epochs

**Epoch**
Time period with a fixed context and needs; characterized by static constraints, design concepts, available technologies, and articulated attributes (Ross 2006)

**Define Epochs**
- Potential Contexts
- Potential Needs

**Construct Eras**
- Epoch Series
- Dynamic Strategies

Discretization of change timeline into short run and long run enables analysis
Allows for rigorous consideration of many possible futures
**Epoch-Era Analysis: Eras**

**Define Epochs**
- Potential Contexts
- Potential Needs

**Construct Eras**
- Epoch Series
- Dynamic Strategies

**Era**
System life with varying contexts and needs, formed as an ordered set of epochs; characterized by varying constraints, design concepts, available technologies, and articulated attributes.

Discretization of change timeline into short run and long run enables analysis.
Allows for analysis of system varying performance over possible futures.
Example Scenario: Two Epochs

Epoch 1
- The Status Quo
- System may degrade over time
- Repair may be possible
- Goal: recover value at \( \min(\text{cost}) \)

Epoch 2
- Tastes change
- Policy changes
- Technology changes
- At day 1, system same, but value discontinuity
- Goal: recover value at \( \min(\text{cost}) \)

Given \( S_{1,b} \):
- Explore \( S_{1,e} \)
- Select \( S_{1,e} \)

If \( S_{2,b} = S_{1,e} \):
- Explore \( S_{2,e} \)
- Select \( S_{2,e} \)

Potential strategies: \( \min(\text{cost}) \), \( \max(\text{value}) \), \( \min(\text{time}) \), combination…

New Customers, Changed Constraints, New Concepts
Epoch Usage Considerations

- An “Epoch” is a mechanism for stringing together short run (Epochs) into long run (Era), simplifying dynamic analysis
- Epochs are defined by system-external “context” changes; timescales are “natural”
- Epochs can be known in advance, or in the moment, deterministic, or probabilistic
- Epoch-Era Analysis can be conducted at any point during system lifecycle, not only conceptual design
- Modularity of Epoch-Era Analysis enhances overall tradespace exploration
- Value (utility) is defined within a given Epoch
- Selection of system end state (goal) within an Epoch is dependent on strategy (min. cost, max. utility, short run vs. long run, etc.)
- System change strategies can be predictive, adaptive, or static
- Multiple strategies for achieving value robustness across an Era

Epoch-Era Analysis can be used for visual communication as well as for quantitative networked tradespace analyses
Achieving Value Robustness

Research suggests two strategies for “Value Robustness”

New Context Drivers • External Constraints • Design Technologies • Value Expectations

1. Passive
   • Choose “clever” designs that remain high value
   • Quantifiable: Pareto Trace number

2. Active
   • Choose changeable designs that can deliver high value when needed
   • Quantifiable: Filtered Outdegree

Value robust designs can deliver value in spite of inevitable context change
Boundary Spanning Activity

- Architect develops possible "scenarios" and "system trajectories"
- Working with analyst, enumerates epochs and eras
- Analyst uses epochs to develop context-dependent models for tradespace exploration
- Software used to generate visualizations and analytically based system trajectories
- Results incorporate perspectives of architects, analysts, and stakeholders

Facilitates discussion and insight
Capturing Information

- Each Epoch has specific quantities associated with it
- Definition of these quantities concretizes a given “context”
- Used as guidance for analyst-developed models

See paper for example
Research Directions

Four further research areas

1. Descriptive research on system context and decision analysis processes
2. Synergies with other time-based methods
3. Validation of method on “real” case study
4. Educational constructs for mindset shift

Example open research question: how can this approach be used to represent or quantify system dynamic properties? (i.e. many of the new “ilities”)

Current research project underway for large scale methodological validation through application to a “real” system
Conclusion

• Epoch-Era Analysis intends to introduce a “natural” value-centric tool for system alternative generation, evaluation, and communication in a dynamic and changing “context”
• Is scalable in application, from qualitative to deeply quantitative
• Useful as boundary object between stakeholders, architects, and analysts
• Force consideration of essential factors for system success: meeting dynamic expectations in changing contexts

Epoch-Era Analysis helps remove the “static-view” bias in current methods by making time a natural dimension for consideration
Thank you!

Any questions?

Thoughts or suggestions? Feel free to email the authors: adamross@mit.edu and rhodes@mit.edu