2010 SEArri Annual Research Summit

Research Topic

“Multi-Epoch Thinking and Considerations for Dynamic Strategies”

Dr. Adam Ross (Lead Research Scientist, SEArri)

October 19, 2010
Cambridge, MA
Massachusetts Institute of Technology
Mismatch of Design with Context

1960’s Paradigm

- CORONA: 30-45 day missions
- 144 spacecraft launched between 1959-1972
  (Wheelon 1997)

Evolution to Current State

- 13+ year design lives
  (geosynchronous orbit)
- Inability to adapt to uncertain future environments, including disturbances

“Our spacecraft, which take 5 to 10 years to build, and then last up to 20 in a static hardware condition, will be configured to solve tomorrow’s problems using yesterday’s technologies.” (Dr. Owen Brown, DARPA Program Manager, 2007)
More than Missed Opportunities: Failures from Context Changes

New competitor/technology changes needs before system completed

Changing contexts can lead a technically sound system to fail
More than Missed Opportunities: Failures from Context Changes

New competitor/technology changes needs before system completed

Adversary timescale shorter than “system” lifecycle

Changing contexts can lead a technically sound system to fail

Changing contexts can have high consequences if systems fail...

Source: Wired Magazine, August 2010
Dynamic System Context

Static View
Structural Metrics

Constraints

Dynamic View
Operational Metrics

System

“Decisional” Value

“Expectations”

“Experienced” Value

“Experience”

**“Decision”, “Experienced”, and “Remembered” Utility from (Kahneman and Tversky 2000)**

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

Static View
- Structural Metrics

Constraints

Dynamic View
- Operational Metrics

System
  - Inputs
  - Outputs

“Decisional” Value
  - “Expectations”

“Experienced” Value
  - “Experience”

System Boundary

“Context”

**“Decision”, “Experienced”, and “Remembered” Utility from (Kahneman and Tversky 2000)**

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

Static View
Structural Metrics

Constraints

Dynamic View
Operational Metrics

“Context”

System Boundary

“Decisional” Value
“Expectations”

System

“Experienced” Value
“Experience”

“Expectations”

“Decision”, “Experienced”, and “Remembered” Utility from (Kahneman and Tversky 2000)

**Discussion of “structural” versus “operational” metrics in (Giachetti et al. 2003)**
• Shift in contexts occur more frequently than typical system development timelines (e.g., budgets, leadership changes, new stakeholder needs)
• Often “time scales” are dictated by programmatic guidelines or arbitrary dates
• Context-derived “natural” time scales may compete with the “artificial” program time scales imposed on a system
  – Alignment of planets (e.g. Mars every 780 days)
  – New administration (e.g. presidential every 4 years)
  – New technology (e.g. Moore’s law: 24 months)
• “Value-centric” time scales focus on dynamic factors that impact the value proposition for the system (e.g. the metrics that define “success”)
Natural Value-Centric Time Scales

- Shift in contexts occur more frequently than typical system development timelines (e.g., budgets, leadership changes, new stakeholder needs)
- Often “time scales” are dictated by programmatic guidelines or arbitrary dates
- Context-derived “natural” time scales may compete with the “artificial” program time scales imposed on a system
  - Alignment of planets (e.g. Mars every 780 days)
  - New administration (e.g. presidential every 4 years)
  - New technology (e.g. Moore’s law: 24 months)
- “Value-centric” time scales focus on dynamic factors that impact the value proposition for the system (e.g. the metrics that define “success”)
Natural Value-Centric Time Scales

- Shift in contexts occur more frequently than typical system development timelines (e.g., budgets, leadership changes, new stakeholder needs)
- Often "time scales" are dictated by programmatic guidelines or arbitrary dates
- Context-derived "natural" time scales may compete with the "artificial" program time scales imposed on a system
  - Alignment of planets (e.g. Mars every 780 days)
  - New administration (e.g. presidential every 4 years)
  - New technology (e.g. Moore’s law: 24 months)
- "Value-centric" time scales focus on dynamic factors that impact the value proposition for the system (e.g. the metrics that define “success”)

System Lifecycle

Now 2012-2050

<table>
<thead>
<tr>
<th>System Transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept Development (3 yrs)</td>
</tr>
<tr>
<td>Design Modifications</td>
</tr>
<tr>
<td>Software Infrastructure Changes</td>
</tr>
</tbody>
</table>
Natural Value-Centric Time Scales

- Shift in contexts occur more frequently than typical system development timelines (e.g., budgets, leadership changes, new stakeholder needs)
- Often "time scales" are dictated by programmatic guidelines or arbitrary dates
- Context-derived "natural time scales" may compete with the "artificial" program time scales imposed on a system
  - Alignment of planets (e.g. Mars every 780 days)
  - New administration (e.g. presidential every 4 years)
  - New technology (e.g. Moore’s law: 24 months)
- "Value-centric" time scales focus on dynamic factors that impact the value proposition for the system (e.g. the metrics that define “success”)

By considering natural value-centric time scales and how systems could respond across the lifecycle, one can begin to anticipate value robust systems that maintain value delivery over time.
An “Epoch” as a Snippet of Time

**Definition of Epoch**
Time period with a fixed context and needs; characterized by static constraints, concepts, available technologies, and articulated expectations

System success depends on the system meeting *expectations* within a given *context* can change! can change!
An “Epoch” as a Snippet of Time

Definition of Epoch
Time period with a fixed context and needs; characterized by static constraints, concepts, available technologies, and articulated expectations

System success depends on the system meeting expectations within a given context

Time

can change!
can change!

t=1  t=2  t=3  t=4  t=

An “Epoch” as a Snippet of Time

**Definition of Epoch**

Time period with a fixed context and needs; characterized by static constraints, concepts, available technologies, and articulated expectations

System success depends on the system meeting *expectations* within a given *context*
An “Epoch” as a Snippet of Time

**Definition of Epoch**
Time period with a fixed context and needs; characterized by static constraints, concepts, available technologies, and articulated expectations.

System success depends on the system meeting *expectations* within a given *context*.

**Epoch-based Thinking**
Using the concept of “epoch” to generate and consider a large number of possible future contexts and needs facing a system, along with short term and long term strategies for maintaining a successful system across epochs.
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.

<table>
<thead>
<tr>
<th>Economic Analysis</th>
<th>Short run</th>
<th>Long run</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed factors</td>
<td>Input factors such as labor, capital, equipment, regulations, knowledge</td>
<td>none</td>
</tr>
<tr>
<td>Variable factors</td>
<td>Quantity demanded, quantity supplied, dollars and time spent</td>
<td>all</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Engineering Analysis</th>
<th>Short run</th>
<th>Long run</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed factors</td>
<td>Design chosen, dollars and time spent, perceived value</td>
<td>none</td>
</tr>
<tr>
<td>Variable factors</td>
<td>all</td>
<td>all</td>
</tr>
</tbody>
</table>

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

Natural analogy from economics to engineering leverages years of methodological validation.
Basic Epoch Shift: System Impact-Response

Baseline Epoch

System Definition

Baseline Epoch
Basic Epoch Shift:
System Impact-Response

Baseline Epoch

System Definition

Baseline Epoch
Basic Epoch Shift:
System Impact-Response

Baseline Epoch

New Epoch

System Definition

Baseline Epoch  New Epoch
Basic Epoch Shift: System Impact-Response

Baseline Epoch  New Epoch
System Definition

Baseline Epoch  New Epoch
Effect
Basic Epoch Shift: System Impact-Response

Baseline Epoch | New Epoch
--- | ---

System Definition

Baseline Epoch → New Epoch
Response(?) → Effect
Basic Epoch Shift: System Impact-Response

Baseline Epoch

System Definition

New Epoch

Baseline Epoch ➔ New Epoch

Response(?) ➔ Effect

Success?
Basic Epoch Shift: System Impact-Response

- What about analyzing across more than just pair-wise epoch shifts?
- How can the technique become generalized or even automated?
Generating Epochs

Many possible contexts and needs may unfold in the future, impacting actual and perceived system utility and cost.

“Epoch-based thinking” can be used to structure anticipatory scenario analysis.

Example triggers for epoch shifts impacting a system:
- Change in political environment
- Entrance of new competitor in market
- Emergence of significant new or changed stakeholder need(s)
- Policy mandate impacting product line, services or operations

Categories of epoch variables can aid in thinking about key changing factors:
- E.g., Resources, Policy, Infrastructure, Technology, End Uses (“Markets”), Competition, etc.
Multi-Epoch Analysis

Epoch Characterization

Epoch set represents potential fixed contexts and needs

Multi-Epoch Analysis

Analysis across large number of epochs reveals “good” designs

Era Construction

Eras represent ordered epoch series for analyzing system evolution strategies

- Cross-epoch analysis
- Within-epoch analysis
- Identification of versatile and changeable designs

For large numbers of designs and epochs, this requires an automated ability to assess epoch shift impacts
Example Metric from Multi-Epoch Analysis: Pareto Trace

Find non-dominated solutions within a given epoch (Pareto Set)
Example Metric from Multi-Epoch Analysis: Pareto Trace

- Find non-dominated solutions within a given epoch (Pareto Set)
- Across many epochs, track number of times solution appears in Pareto Set
Example Metric from Multi-Epoch Analysis: Pareto Trace

Find non-dominated solutions within a given epoch (Pareto Set)

Across many epochs, track number of times solution appears in Pareto Set

Identify designs with high Pareto Trace for further investigation

e.g. “design 3435” is in 67% of Pareto Sets

Higher Pareto Trace designs are more passively value robust
Performing Pareto Tracing to Find Epoch Shift Robust Solutions

Using multi-epoch analysis, one can perform anticipatory exploration of possible preferences for a system
– Answering “what if” questions on needs…

• What if you don’t elicit the “right” attribute priorities?
• What if you don’t elicit all of the “right” attributes?
• What if you don’t elicit the “right” utility curve shape?
• What if you don’t use the “right” utility aggregating function?
• What if a second decision maker enters the mix?
• …

– Finding designs in common across varying stakeholder sets and needs over time

Pareto Trace is a metric of passive value robustness across epoch variations

While Pareto Trace is a useful screening metric for “passive” strategies, what about changing systems over time to achieve value?
Research suggests two main strategies for “Value Robustness”

New Context Drivers

- External Constraints
- Design Technologies
- Value Expectations

1. Passive
   - Choose “versatile” or “robust” designs that remain high value
   - Ex. Metric: Pareto Trace number

2. Active
   - Choose “changeable” designs that can deliver high value when needed
   - Ex. Metric: Filtered Outdegree

Value robust designs can deliver value in spite of inevitable context change
Using Epochs for Proposing Time-Based Strategies

Discretization of change timeline into short run and long run enables analysis

Epoch:
Time period with a fixed context; characterized by static constraints, design concepts, available technologies, and articulated needs

Era:
Time-ordered sequence of epochs

Needs (performance, expectations)

- System
- Expectation 1
- Expectation 2
- Expectation 3
- Expectation 4
- Changed System
- Unchanged System

Legend
- System
- System Trajectory
- Expectations

In order to pursue dynamic strategies, a system must have temporal properties, i.e., "-ilities" such as flexibility, evolvability, or survivability

(Ross and Rhodes 2008)
Defining Some “ilities”

<table>
<thead>
<tr>
<th>ility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>robustness</td>
<td>ability of a system to maintain its level and set of specification parameters in the context of changing system external and internal forces</td>
</tr>
<tr>
<td>versatility</td>
<td>ability of a system to satisfy diverse needs for the system without having to change form (measure of latent value)</td>
</tr>
<tr>
<td>changeability</td>
<td>ability of a system to alter its form—and consequently possibly its function—at an acceptable level of resource expenditure</td>
</tr>
<tr>
<td>flexibility</td>
<td>ability of a system to be changed by a system-external change agent</td>
</tr>
<tr>
<td>adaptability</td>
<td>ability of a system to be changed by a system-internal change agent</td>
</tr>
<tr>
<td>scalability</td>
<td>ability of a system to change the current level of a system specification parameter</td>
</tr>
<tr>
<td>modifiability</td>
<td>ability of a system to change the current set of system specification parameters</td>
</tr>
<tr>
<td>survivability</td>
<td>ability of a system to minimize the impact of a finite duration disturbance on value delivery</td>
</tr>
<tr>
<td>value robustness</td>
<td>maintaining value delivery in spite of changes in needs or context</td>
</tr>
</tbody>
</table>

Defining, characterizing, and quantifying “ilities” is an active area of research, necessary in order to specify and validate such system properties in practice.
Example: Defining Adaptability

<table>
<thead>
<tr>
<th>changeability</th>
<th>ability of a system to be altered, by intention, in form—and consequently possibly in function—at an acceptable level of resource expenditure</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexibility</td>
<td>ability of a system to be altered by a system-external change agent</td>
</tr>
<tr>
<td>adaptability</td>
<td>ability of a system to be altered by a system-internal change agent</td>
</tr>
</tbody>
</table>

An adaptable system is one that can intentionally alter itself, typically in response to a perturbation (such as a change in context), in order to improve its value.

In order to develop rigorous techniques to assess and design for adaptability, this definition must be deconstructed into a useful framework.
Framework for Assessing and Designing for Changeability

Change pathway: Perturbation-Agent-Mechanism-Effect

At least two questions can be asked regarding degree of changeability

1. Can a system be changed or change itself?
   - Capability question; pursue structural and operational strategies
   - Key metrics: number of destination end states, time/cost to achieve change

2. Does the change result in a “better” system?
   - Value question; analyze context-dependent performance and perceptions
   - Key metrics: utility loss/gain over time, aggregate value delivery/availability
Framework Example

Change pathway: Perturbation-Agent-Mechanism-Effect

- **Perturbation**
  - Δ technology
  - Δ business model
  - Δ enemy

- **System**
  - Sat. constellation
  - Aircraft fleet
  - Training system

- **Agent**
  - Program manager
  - Autonomous software
  - Warfighter

- **Mechanism**
  - Swap payloads
  - Redefine operations
  - Combine assets

- **Effect**
  - New capability
  - Enhanced survivability
  - Cost-efficiency gains

Framework helps to structure thinking about changeability, including creative generation of strategies, as well as quantification.
“Dynamicism”
Matching Change with Change

- Contexts and needs for systems will inevitably shift, especially over long-lived systems
- Timescales for change are often out of a designer’s control
- Multi-epoch thinking is a structured means for considering an array of possible short-run futures
- Eras can be used to characterize path-dependence of the context in the long run

Determining the best strategy, both short run and long run, is a difficult temporal problem, fraught with uncertainty.
Considerations for Dynamic Strategies

- Developing dynamic system strategies requires consideration of both the future context and needs, as well as matching system dynamic capabilities (i.e., “ilities”)
- The value of having an “ility” such as adaptability is context-dependent, however analysis can be used to generate insights
- Oftentimes decisions for incorporating an “ility” into a system occurs prior to the recognition of need for such a capability
  - Costs: design, carrying, “utility-loss,” and execution
  - Benefits: at later point in time can alter system to higher value state
- Development of rigorous strategies an on-going area of research

Determining the best strategy, both short run and long run, is a difficult temporal problem, fraught with uncertainty.

Recent research seeks to better characterize the problem and suggest design principles for increasing the likelihood of realizing value robust systems.