A Taxonomy of Perturbations: Determining the Ways That Systems Lose Value

IEEE International Systems Conference
March 21, 2012

Brian Mekdeci, PhD Candidate
Dr. Adam M. Ross
Dr. Donna H. Rhodes
Prof. Daniel E. Hastings

Massachusetts Institute of Technology
Cambridge, MA
Value Robustness of Systems

- Engineered systems are designed to deliver value for stakeholders.
  - Value being some utility or benefit to the stakeholders, at some cost
- Systems *fail* when they no longer produce an acceptable value to stakeholders, during some specified period.
  - Failures of large, complex systems have been prominent in recent news:
    - Japanese nuclear power plants
    - Sony PlayStation Network (PSN)
    - Amazon’s Elastic Compute Cloud

Thus, system architects need to understand *what* causes systems to fail.
Example scenario: Fire caused by lightning

• Suppose a structure is struck by lightning, ignites, and burns down

- Perturbation: Unintended state change of a system’s form, operations or context, which could jeopardize value delivery
- Disruption: Instantaneous, discontinuous perturbation (e.g. lightning)
- Disturbance: Finite duration, continuous perturbation (e.g. fire)

- Threat: An external set of conditions that exist which may cause a perturbation, but hasn’t impacted value delivery, yet. (e.g. thunderstorm)

- Hazard: An internal set of conditions inside a system that can cause a perturbation (e.g. flammable building materials).
**Survivability and Value Robustness**

**Survivability**: The ability of systems to prevent, mitigate and recover from value delivery reduction as a result of some perturbations.

**Three Types of Survivability:**

I. Prevention  
II. Mitigation  
III. Recovery

---

**Example Scenario: Automobile accident**  
Suppose an automobile manufacture wants to make its car “survivable” in the event of a collision.  
- Active Type II survivability design principles are not applicable  
  - No time to react  
- Death or injury may be unrecoverable

Need to understand the nature of the perturbation to be survivable against them
Making Systems Survivable

Example scenario:
- Exhausted pilot flies through thunderstorm
  - Rain reduces visibility
  - Too tired to notice low altitude
- A wing gets damaged by clipping tower
- Plane spirals out of control, crashes, explodes

How can system architects make a plane survivable in such a scenario?

- Wing clip
  - Land safely with a damaged wing
    - Does the concept of operations include the fact that the weather is bad and the pilot is tired?
    - Could have done more damage
- Low visibility
  - Include windshield wipers
    - Similar problem at nighttime
    - May not matter if the pilot is tired

Characterizing Perturbations, Threats & Hazards

**Nature**
- How does the disturbance impact the system?

**Origin**
- Internal or external to the system
- For many SoS, the lines are blurred.

**Intent**
- Is there an intent, by some entity, to cause this disturbance?

**Length of Impact**
- How long is the duration of the disturbance?
- Does the original context resume?

Effectiveness of a design principle will be strongly dependent on characteristics of the perturbations, threats & hazards
Determining a Suitable Taxonomy

- Classifying perturbations by “type” is great if system architects want to focus on very specific perturbations and ignore others
  - E-commerce sites may want to focus on hacker attacks, while yogurt manufacturers may choose to ignore them
- However, dismissing entire classes of perturbations without analysis is risky
  - Assuming we know what to expect (“known unknowns”)
  - Some of the biggest system failures were the result of events that system architects never considered
    - 9/11 attacks
    - 2003 Northeast Blackout
- A solution to a particular problem, may be the solution to another problem as well
  - An authentication procedure can not only protect against hacker attacks, but also against unintentional actions by legitimate users.
Fault Tree Analysis

- Top-down approach (Fenelon et al., 1994)
- Uses Boolean logic to determine cause of a single failure (effect)
- Deductive approach that often does not discover multiple effects of a single cause

http://www.emeraldinsight.com/journals.htm?articleid=841186andshow=html
FMEA/FMECA

- Failure Mode Effects (and Criticality) Analysis
- Bottom-up approach
- Addresses “loss of an intended function of a device” i.e. component / capability failures, not operational / human failures (Langeford, 1995) (FAA, 2004)
- Very linear
  - Does not show multiple causes and effects or complex relationships well

<table>
<thead>
<tr>
<th>Item Identification</th>
<th>Function</th>
<th>Failure Mode</th>
<th>Failure Cause</th>
<th>Component or Functional Assembly</th>
<th>Next Higher Assembly</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch</td>
<td>Initiates Motor Power Function</td>
<td>Fails to Open</td>
<td>Release Spring Failure, Contacts Fused</td>
<td>None</td>
<td>Maintains Energy to Circuit Relay</td>
<td>Maintains Energy to Pwr Circuit Through Relay</td>
</tr>
<tr>
<td>Battery #2 (Relay Circuit)</td>
<td>Provides Relay Voltage</td>
<td>Fails to Provide Adequate Power</td>
<td>Depleted Battery, Plates Shorted</td>
<td>None</td>
<td>Battery Gets Hot and Depletes</td>
<td>Fails to Operate Relay Circuit</td>
</tr>
<tr>
<td>Relay Relay Coil</td>
<td>Closes Relay Contacts When Energized</td>
<td>Coil Fails to Produce EMF</td>
<td>Coil Shorted or Open</td>
<td>Does Not Close Relay Contacts</td>
<td>Does Not Energize Pwr Circuit</td>
<td>System Fails to Operate</td>
</tr>
<tr>
<td>Relay Contacts</td>
<td>Energizes and De-Energizes Pwr Circuit</td>
<td>Fails to Open</td>
<td>Contacts Fused</td>
<td>None</td>
<td>Maintains Energy to Motor</td>
<td>Overheated Pwr Circuit Wire if Motor is Shorted and Circuit Breaker Fails to Open</td>
</tr>
<tr>
<td>Motor</td>
<td>Provides Desired Mechanical Event</td>
<td>Fails to Operate</td>
<td>Motor Shorted</td>
<td>Motor Overheats</td>
<td>High Current in Pwr Circuit</td>
<td>Overheated Pwr Circuit Wire if Circuit Breaker Fails to Open and Switch or Relay Fails</td>
</tr>
<tr>
<td>Circuit Breaker</td>
<td>Provides Pwr Circuit Fusing</td>
<td>Fails to Open</td>
<td>Contacts Fused</td>
<td>None</td>
<td>Maintains Pwr to Motor if Relay Contacts are Closed</td>
<td>Maintains Energy to Motor</td>
</tr>
<tr>
<td>Battery #1 (Pwr Circuit)</td>
<td>Provides Motor Voltage</td>
<td>Fails to Provide Adequate Power</td>
<td>Depleted Battery, Plates Shorted</td>
<td>None</td>
<td>Battery Gets Hot and Depletes</td>
<td>None</td>
</tr>
</tbody>
</table>

http://www.fmeainfocentre.com/examples.htm
Everything that causes a reduction in value delivery has at least one cause, and at least one effect.

Each cause is a set of conditions that led to the perturbation.

The effects are the change in context and/or system that are a direct result of the perturbation.

Exactly what caused a perturbation, may not be known, neither what effect(s) it has. These can be called *unknown unknowns*.
Multiple Causes, Multiple Effects

Many perturbations have multiple causes and/or multiple effects

- Not possible to make system survivable against all perturbations
- Constraints:
  - Budget
  - Time
  - Resources
- Qualitative characteristics of perturbations
  - Difficult to quantify
  - Difficult to model

Separating perturbations into cause and effect provides system architects with a qualitative way to prioritize causes / effects
Cascading Failures

The effects of some perturbations, become the cause of others, in what’s known as a cascading failure.

Systemic Risk – The risk that a cascading failure will result from entities being too interconnected with each other.
An effect of “bad weather” is blurry images due to precipitation buildup on lenses.

- **Cause** – Precipitation on lenses
  - Type I – Prevention by sheltering the lens
- **Effect** – Blurry images
  - Type II – Mitigation perform image processing

Separating perturbations into cause and effect allow system architects to focus on what they can affect and what they can’t.

http://www.impactlab.net/2008/04/14/laser-used-to-trigger-lightning-in-a-thunderstorm/
Cause and Effect Mapping

**Purpose:**
- To highlight the complex, non-linear relationship between causes and effects of perturbations

**Method:**
- Only potential perturbations that can affect the system (or for which the system can influence) are considered
- Start with an effect, determine immediate cause(s), see what other immediate effects result.
- Link existing cause/effects to each other, if appropriate
Cause and Effect Mapping

Highlights:

• Shows multiple causes / multiple effects
  • Some perturbations are more connected than others
• Exposes cascading failures
• Encourages system architects to recognize relationships that may not have been obvious
• General, rather than specific
  • Allows similar perturbations to benefit from same design principles / strategies
• Useful for broad analysis
  • FTA, FMEA/FMECA useful for specific perturbations
Commonalities Between Perturbations

Main Effects:
- Capability loss
- Capability degradation
- Change in mode of operation
- Cost increase
- Change in stakeholder expectation

Focusing on the main effects may yield the most useful value robustness strategies against unknown unknowns.
## Example List of Perturbations, Causes, Effects and Solutions

<table>
<thead>
<tr>
<th>Perturbation Example</th>
<th>Type</th>
<th>Immediate Effect</th>
<th>Main Effects</th>
<th>Causes of Perturbation</th>
<th>Survivability Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lightning Strike</td>
<td>Disruption</td>
<td>Physical damage to components</td>
<td>Capability loss, capability degradation</td>
<td>Context change (weather)</td>
<td>Decrease cross-sectional area, divert lightning away (e.g., lightning rod)</td>
</tr>
<tr>
<td>Crash</td>
<td>Disruption</td>
<td>Physical damage to components</td>
<td>Capability loss, capability degradation</td>
<td>Collision (caused by operator error, context change, diminished situational awareness)</td>
<td>Decrease cross-sectional area, increase maneuverability, increase situational awareness</td>
</tr>
<tr>
<td>Fuel Price Increase</td>
<td>Disruption</td>
<td>Cost increase</td>
<td>Cost Increase</td>
<td>Resource scarcity, mode of operation change</td>
<td>Store excess resource when not scarce, change to alternate resource</td>
</tr>
<tr>
<td>Stakeholder Changes Mind About Pollution</td>
<td>Disruption</td>
<td>Capability loss</td>
<td>Change in stakeholder expectations</td>
<td>Context change (stakeholder)</td>
<td>Change components / mode of operation accordingly.</td>
</tr>
<tr>
<td>Operator Gives Wrong Command to Machine</td>
<td>Disruption</td>
<td>Capability degradation</td>
<td>Change in mode of operation</td>
<td>Context change (weather, bad working conditions), workload exceeds component capacity</td>
<td>Increase capacity (increase operators, increase automation), increase training,</td>
</tr>
</tbody>
</table>
Discussion and Future Work

• Eventual goal is to develop design principles that will guide system architects to produce systems that provide value no matter what
• Working towards that goal by
  – Clarifying differences between disturbances and disruptions, so system architects can apply appropriate design principles
  – Showing how using causal chains and working backwards from value impact, systems architects can begin to determine where to intervene
  – Showing that by using cause and effect mapping, general categories of effects can be useful as a taxonomic basis, especially for dealing with known unknowns and potential unknown unknowns

Future Work:
• Apply cause and effect mapping to case studies
  – E.g. Maritime security SoS
• Use cause and effect mapping (along with other analysis methodologies) to develop survivability / value robustness strategies
• Evaluate, refine cause and effect mapping accordingly
End of Presentation

Thank you!