Design Principles for the Survivability of Systems of Systems

Brian Mekdeci, Ph.D. in Engineering Systems (expected in 2012)

Committee: Prof. D Hastings, chair; Dr. Donna Rhodes; Dr. Adam Ross; Prof. Dan Frey

Systems Engineering Advancement Research Initiative

Massachusetts Institute of Technology
Engineering Systems Division

Biography

Brian completed a B.A.Sc. (2002) and a M.A.Sc. (2005) in Systems Design Engineering at the University of Waterloo in Canada. Afterwards, he worked at CDL Systems Ltd in Calgary, Alberta as a Systems Engineer in charge of designing ground control station software for unmanned aerial vehicles. Currently, Brian is researching survivability of systems of systems as part of his doctoral studies at the Massachusetts Institute of Technology.

mekdeci@mit.edu

Related Publications

Mekdeci, B., Ross, A.M., Rhodes, D.H., and Hastings, D.E., "System Architecture Pliability and Trading Operations in Tradespace Exploration," 5th Annual IEEE Systems Conference, Montreal, Canada, April 2011.

Mekdeci, B., Ross, A.M., Rhodes, D.H., and Hastings, D.E., "Examining Survivability of Systems of Systems," INCOSE International Symposium 2011, Denver, CO, June 2011.

Motivation

Failures of large, complex systems have been prominent in recent news:

- Japanese nuclear power plants
- Sony PlayStation Network (PSN)
- Amazon's Elastic Compute Cloud (EC2)

Stakeholders want systems that provide acceptable value

- Over long life cycles
- Subject to various disturbances / context changes
- Which balances performance, cost, and risk

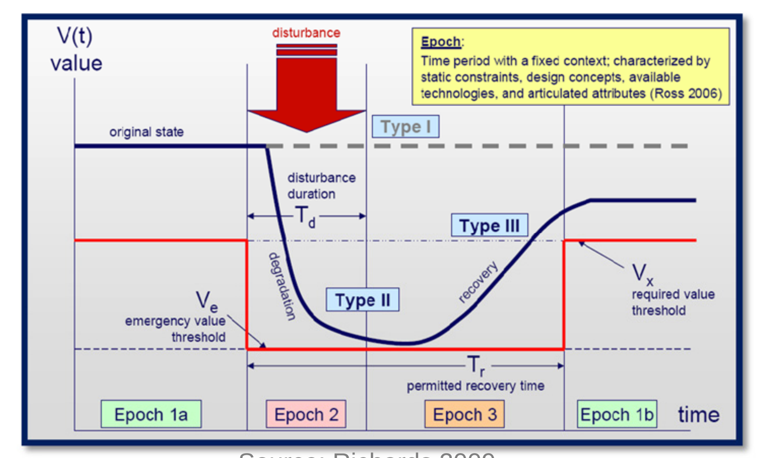
Delivering value is particularly difficult for systems of systems (SoS) that have diverse stakeholders (Ellison & Woody, 2007), due to variation in:

- Needs & expectations
- Risk management strategies
- Available resources

As traditional systems get interconnected and overall complexity increases, designers, architects and decision makers need design principles that will enable and enhance SoS survivability

Perturbations & Survivability

- *Perturbations* are changes in the system or the context, which may impact a system's ability to provide value
- *Epochs* are time periods with a fixed context; characterized by static design concepts, constraints, technologies and stakeholder needs (Ross, 2006)
- *Disruptions* are instantaneous events that cause an epoch change or system change.
- *Disturbance* is an epoch itself where the system's value delivery can be degraded beyond it's normal threshold.
- Disruptions often cause disturbances, which can cause further perturbations, in a cause-impact chain.
- A system is survivable if it can continue to provide an acceptable level of value after a disturbance or disruption.
- There are three ways to achieve survivability:
  - I. **Susceptibility reduction** - Making a disturbance/disruption less likely to impact the system
  - II. **Vulnerability reduction** - Reducing the degradation in system performance due to a disturbance
  - III. **Resilience enhancement** - Increasing the system's ability to recover



Source: Richards 2009

	Disruption	Disturbance
Length of Impact	Instant	Short
Examples	Lightning strike, component failure, policy change, terrorist attack, earthquake,	Flying with no engine, fire, rain, turbulence, high fuel prices, political crisis, sickness, flooding

Characterizing Perturbations

Perturbation Example	Type	Immediate Effect	Causes of Perturbation	Location of Cause	Intent of Cause
Aircraft struck by lightning	Disruption	Change in form, component degradation	System is in bad weather, form acts like lightening rod	External	No
Crash between system and other mobile entity, crash between system and environment	Disruption	Change in form, component degradation	Physical contact	External	No
High winds creating turbulence	Disturbance	Change in form, component damage, change in mode of operation	System in area of high winds, aerodynamics of aircraft	External	No
Precipitation builds on lenses	Disturbance	Context degradation	Lens in contact with precipitation	External	No
Fuel price increase	Disruption	Context degradation	Consumption of external resource	External	No
Environmental ozone regulation makes component obsolete	Disruption	Change in form / mode of operation	Component is subject to environmental regulations, component produces hazardous substances	External	No
Operator gives wrong command to machine	Disruption	Change in mode of operation	Machines not fully automated (require operators), fatigue, poor training, random chance, sabotage operator allowed to make an error, local information	Internal	No
Communication interference	Disturbance	Change in mode of operation	Jamming, unintentional broadcast, precipitation between sender and receiver	Internal	No
Heat from GPU interferes with CPU	Disturbance	Change in form, component degradation, change in mode of operation	Unintentional interconnections (physical proximity between components)	Internal	No
Missile strikes aircraft	Disruption	Change in form, component degradation.	Physical proximity, aircraft has large cross-sectional area, enemy has capability	External	Yes
Friendly artillery unit withdraws from SoS	Disruption	Change in form	Component has operational / managerial independence	Internal / External	Yes
DDOS attack	Disturbance	Capacity exceeded	Server accepts unsecure client requests	External	Yes
Bacterial infection	Disturbance	Mode of operation change, change in form / damage, resources consumed	Open system exchanges matter with environment, system has resources that outside entities want to consume	External	No
Random component failure	Disruption	Change in form / damage	Lack of resources, poor maintenance, random chance	Internal	No
Miscommunication between components	Disruption and/or disturbance	Change in mode of operation	Components are explicitly interconnected, local knowledge, protocol errors, poor connection	External	Yes / No

Pliability in System Architectures

Pliability: *The ability to be easily "bent" without breaking*

The ***pliable range*** of a SA is the set of allowable values the SA parameters can take (i.e., the "guaranteed" set of allowable system choices)

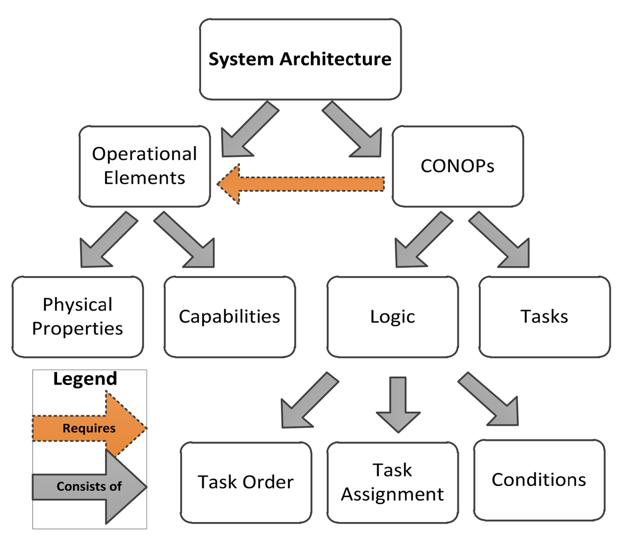
- Sets "bounds" on the allowable system instances

The ***pliability*** of a system is the ability of the system to change from one instance of a SA into another instance of the same SA

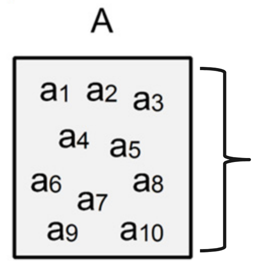
- Changes occur at the parameters
- If the parameter was pliable, then SA remains the same

Pliability relies on two conditions

1. The new instance is part of the original SA (i.e., the parameter values are allowed as defined in the pliable range)
2. The transition is possible



System Architecture
A



Hypothesis: Systems that are more pliable than others, have latent value due to their ability to transition to other validated instances. The larger the pliable range of a SA, the more survivable its systems will be.

Maritime Security Case Study

Key SoS issues:

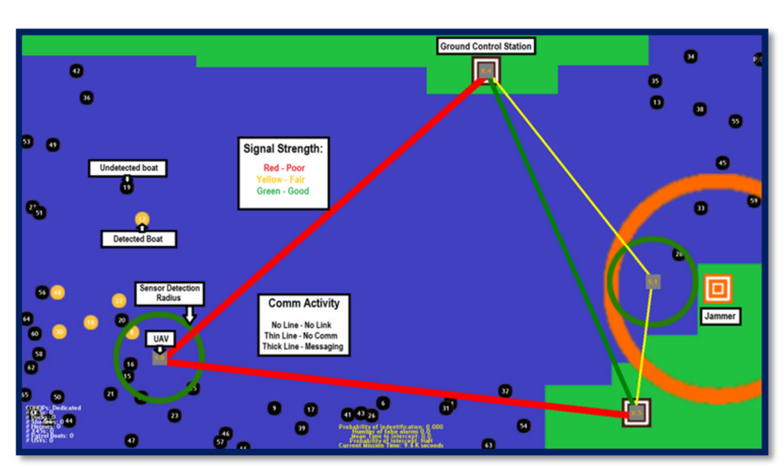
- Component systems geographically separated must share local knowledge
- Dynamic configuration may remove functionality/capacity, exceed bandwidth, interfere with CONOPs

Form

- Composition:
  - All unmanned
  - Mix of manned/unmanned
  - Number of operators
  - Number of ground control
- Technology
  - RF or EO/ IR sensors

CONOPs

- Roles:
  - Distinct / overlapping
  - Yes / no
- Take off and landing
  - Patrol boats /mainland
- Interception
  - UCAV / patrol boats



Discrete Event Simulation (DES)

- Agent based modeling allows for key SoS properties to emerge.
- Allows for real-time visualization, for model verification and CONOPs planning
- Integrates with epoch-era model and generates the performance data necessary for tradespace evaluations of many designs and contexts.

Emerging Survivability Design Principles

Redirection	•Type I survivability design principle •Divert disturbances away from vulnerable components
Defensive Posture	•Type I survivability design principle •Be liberal in what you receive, and conservative in what you send •Taken from Postel's Robustness Principle (1981) •Cited as being one of the main reasons why the Internet has been so robust
Stable Intermediate Forms	•Type II survivability design principle •Explicitly design for evolutionary development •System will produce value, with difference components / CONOPs •Allows "fall back state" in case of disturbance
Adaptation	•Type III survivability design principle •System deliberately changes value function by altering its form and/or CONOPs in the presence of the disturbance

Next Steps

- Use DES to experiment with different designs / contexts / perturbations
- Review case studies for additional disturbances / disruptions. and survivability strategies
- Develop taxonomy of system characteristics
- Develop taxonomy of perturbations
- Develop survivability design principles that relate system characteristics to their effectives in surviving various perturbations.

For more information, please visit: http://seari.mit.edu