# System Architecture Pliability and Trading Operations in Tradespace Exploration


Brian Mekdeci

Adam M. Ross, Donna H. Rhodes, Daniel E. Hastings

Massachusetts Institute of Technology



IEEE International Systems Conference

April 4-7, 2011

Montreal, Canada

# Motivating Problem

Design a complex system that:

- Operates in a variety of environments and contexts
- Satisfies numerous mission objectives
- Operates over a long life cycle

http://atlas.nrcan.gc.ca/site/english/maps/environment/seaice/sar_satellite.jpg

http://strangehorizons.blogspot.com/2011/01/age-of-uav.html

http://media-1.web.britannica.com/eb-media/97/99697-004-DA347454.jpg

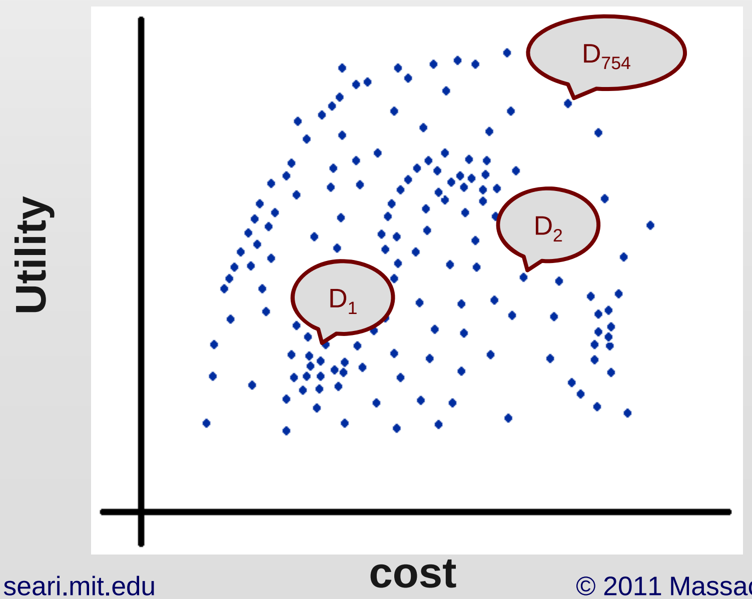
# Tradespace Exploration

- "Design by shopping" (Balling, 1999)
- Allows decision makers to compare designs visually (Stump et al., 2004), (Ross et al., 2004)

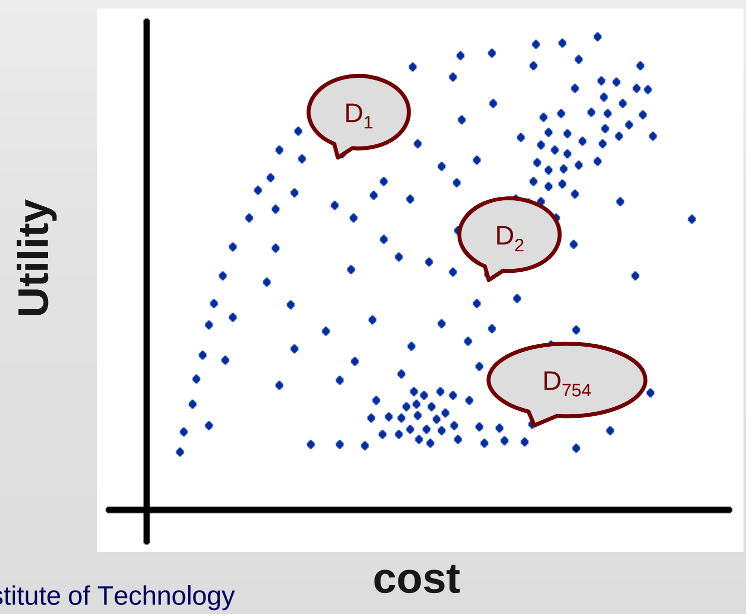
## Sample Tradespace Exploration

$$D_1 = \begin{bmatrix} 4\ UAVs \\ EO\ Payload \\ \vdots \\ Rotary\ Wing \end{bmatrix} D_2 = \begin{bmatrix} 6\ UAVs \\ EO\ Payload \\ \vdots \\ Rotary\ Wing \end{bmatrix} \dots\ D_{754} = \begin{bmatrix} 4\ UAVs \\ IR\ Payload \\ \vdots \\ Fixed\ Wing \end{bmatrix}$$

**Target Detection**




**Area Coverage**



# Limitations of Tradespace Evaluations

- Operational variables are important, particularly for systems of systems
  - (Mekdeci & Cummings, 2009), (Ross, 2006), (Nickel, 2010)

  but are often not included
    - Vendors and component availability often drive physical design options
      - Operational variables may not be as obvious
    - Changing operations often involve substantial changes to models and simulations

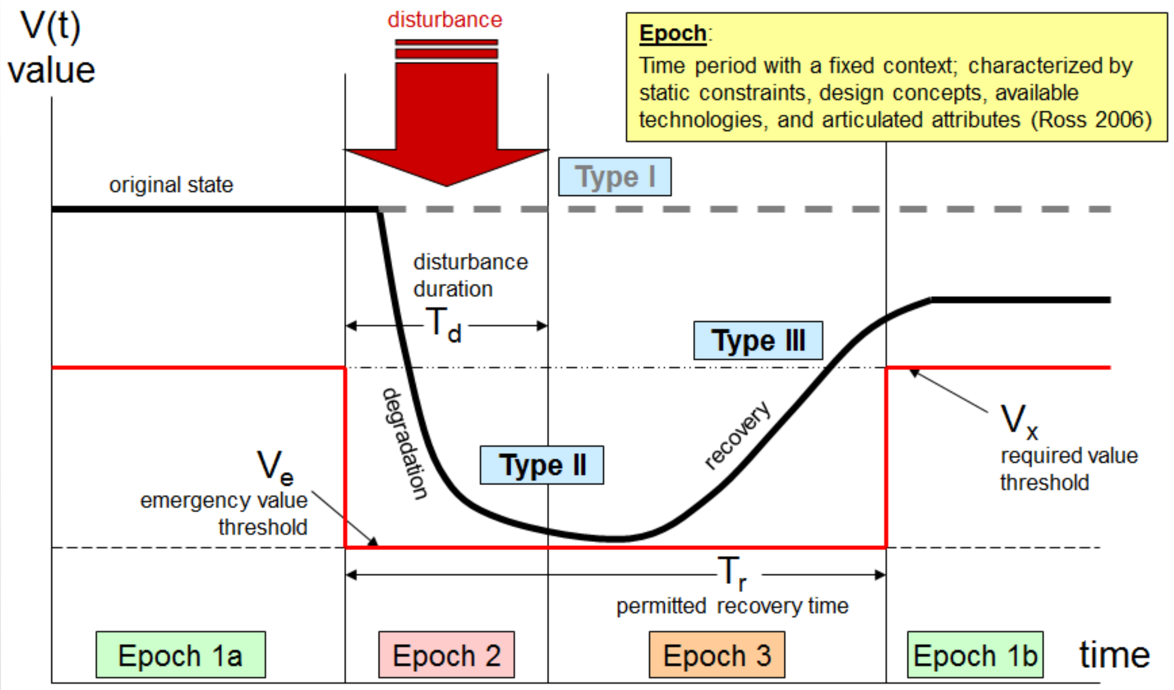
**How can operational choices be explicitly incorporated into design option evaluations?**

- Tradespace can become too large for exhaustive exploration
  - A dozen variables, each with 3 or 4 options = millions of different designs
  - Can require expensive modeling & simulation

**How can the space of possible system designs be intelligently sampled into a reasonable set of design options?**

# Survivability & Robustness

- Three types of Survivability (Richards, 2009)
  - I. Decreasing susceptibility to disturbances
  - II. Decreasing vulnerability to disturbances
  - III. Increasing resilience from disturbances
- If disturbance induces permanent context change, robustness is the issue



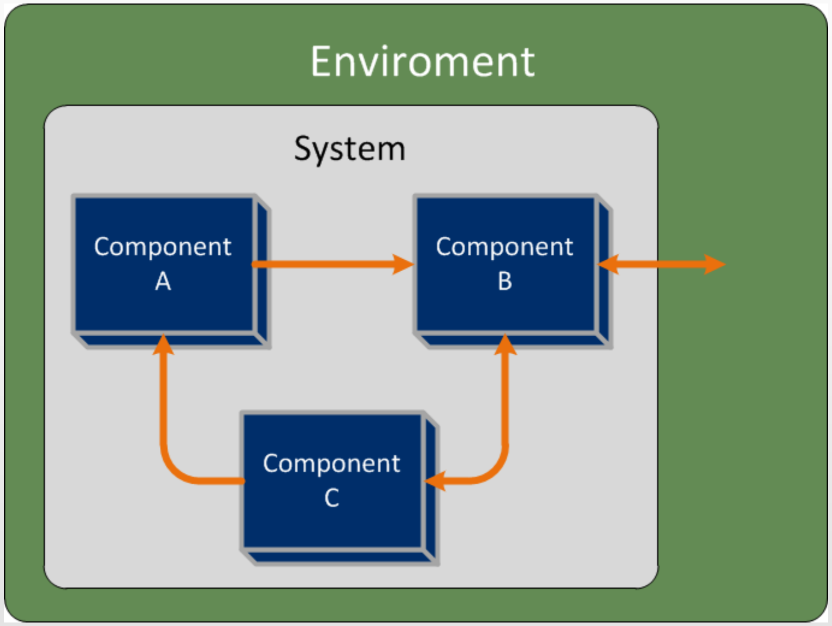

Richards (2009)

**How can system designers, architects and decision makers increase system survivability and robustness?**

# Including Operational Variables

- Systems consist of components, that interact with each other and their environment, to do *something.*
- "Point designs" typically focused on physical components

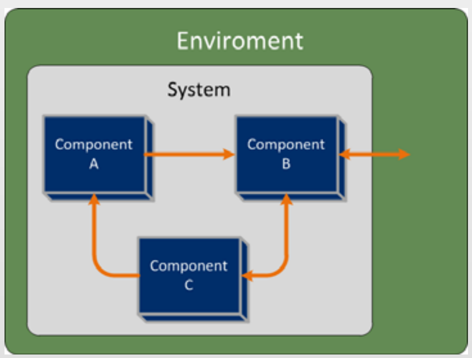
$$D_1 = \begin{bmatrix} 4\ UAVs \\ EO\ Payload \\ \vdots \\ Rotary\ Wing \end{bmatrix}$$




- The "design" of a system should explicitly include
  - A description of the components
  - A description of how the components interact with each other and the environment, to do *something*.
    - Concept of Operations (CONOPs)

# System Architecture

- A *system architecture* (SA), is a collection of operational elements (components) and associated concept of operations, whose instances provide some value, for a particular context.
- A system architecture is a design, whereas a *system* is an actual realization (instance) of a particular SA.
  - e.g.
    - The Nimitz-class aircraft carrier is a SA
    - The USS Ronald Reagan is a system

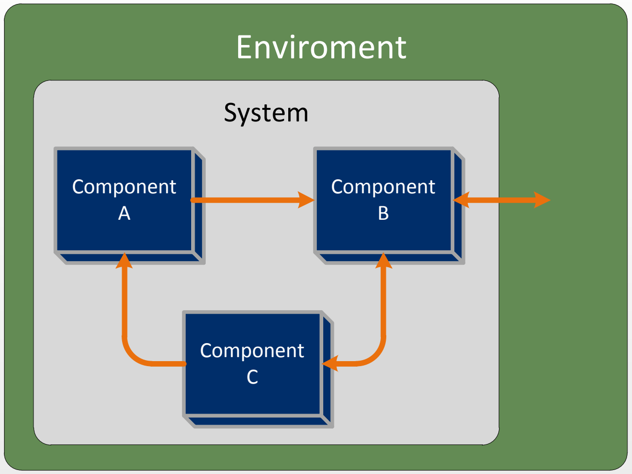


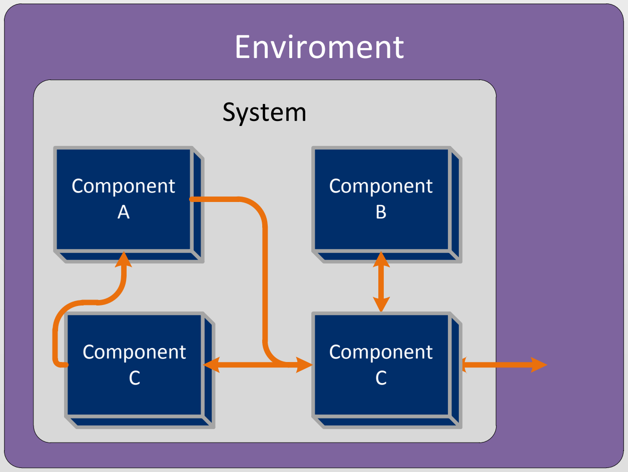

http://www.danzfamily.com/archives/2004/07/uss_ronald_reag.php

SAs in tradespace evaluations can explicitly include operational choices by varying the CONOPs in addition to varying operational elements

# Dynamic Systems

- What sort of SA do complex systems need?
- SAs should be designed such that their instances produce acceptable value for any context that they system may be operated in.
  - Static systems are **not** likely to do this
- Systems need to be
  - Changeable
  - Flexible
  - Adaptable
  - Modifiable
  - Scalable



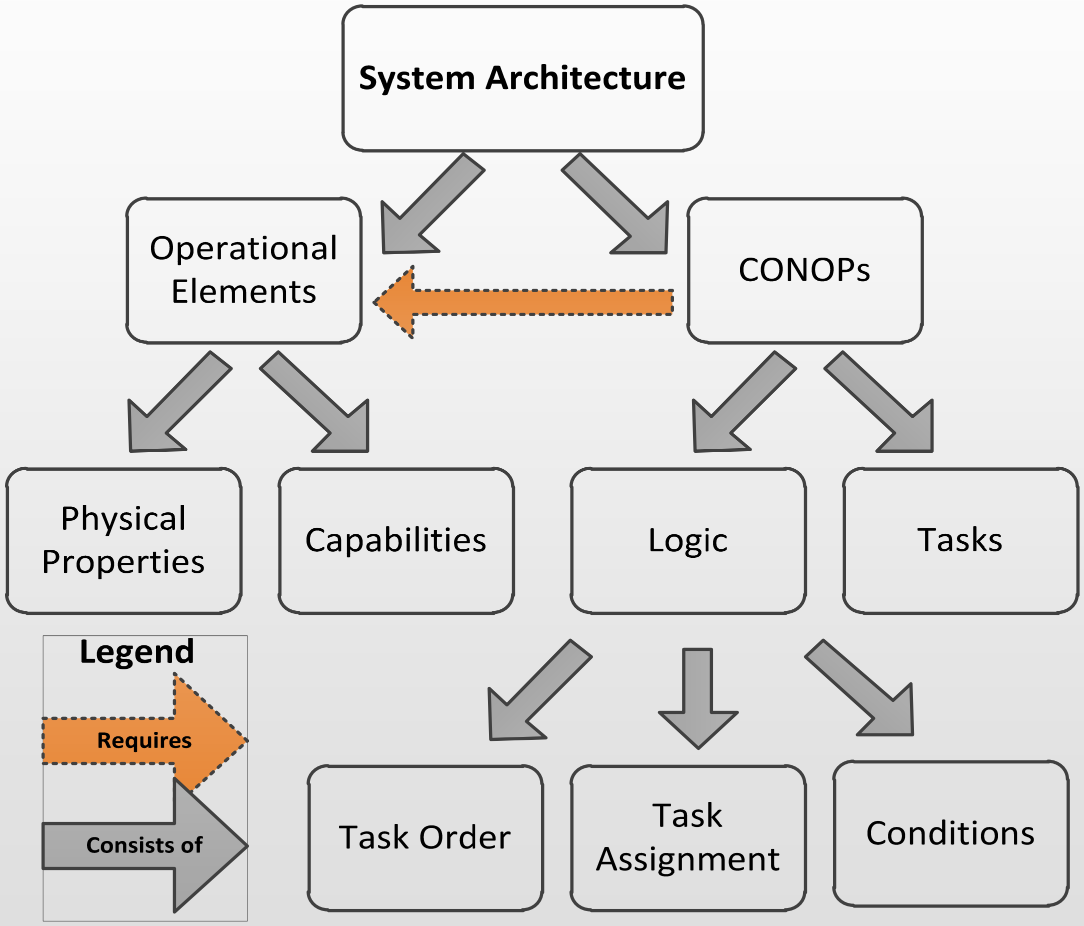
# Dynamic Systems

- Sometimes change is ***not*** intentional
  - Disturbances
    - Manmade
    - Natural
  - Systems need to be survivable
- Typically, the longer the lifecycle of the system, the more likely the context will change or a disturbance will occur


http://india-spicy.blogspot.com/2011/03/japan-nuclear-crisis-2011-letest.html

How do we know that when a system changes (intentionally or not), it will still provide value?

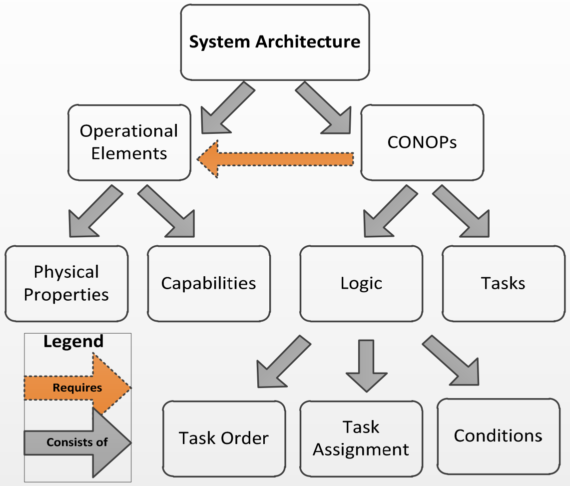
# Parameters of a System Architecture

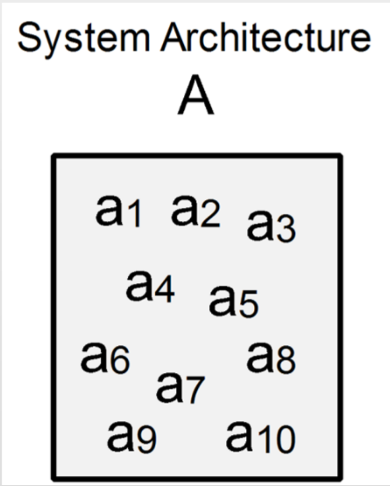



- A parameter is any part of a SA
  - The physical properties & capabilities that describe the OEs
  - The tasks, task order, task assignments and conditions that describe the CONOPs
- All parameters are either fixed or not
  - The system architect decides what parameters are allowed to change, and the range, since he/she validates that design

# Pliability

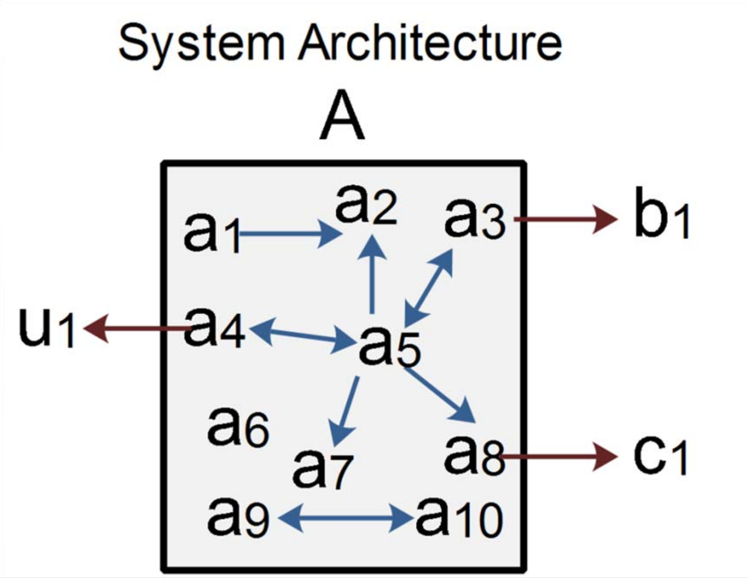
- Definition of Pliability (WordNet 3.0, Merriam-Webster):
  - *The ability to be easily bent without breaking*
- The ***pliable range*** of a SA is the set of allowable values that the parameters can take.
  - Sets the "bounds" on the systems.
- The ***pliability*** of a system is the ability of the 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 the system has maintained the same SA.
  - Otherwise, the system has transitioned to a new SA.
- Pliability relies on two conditions
  - The new instance is part of the original SA (i.e. the parameter changes are allowed as defined in the pliable range)
  - The transition is possible



# Pliability, Changeability and Survivability

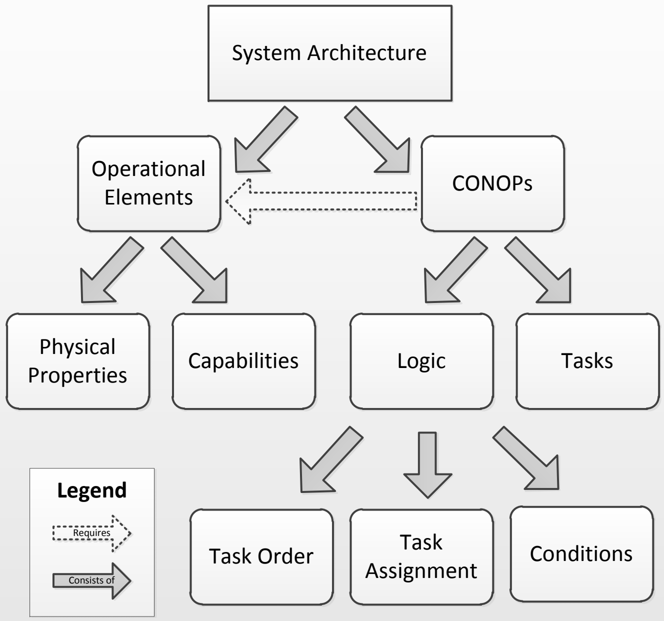
- For transition to be possible
  - The stakeholders must allow it
    - Changeability (Ross, 2006)
      - Not the same as pliability
- Increasing pliable range of a SA may increase chances of survivability
  - The more instances of a SA
    - the more likely an unintentional transition will remain within the pliable range
    - The more likely the system can transition back (through repair or replacement) from a degraded state.




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

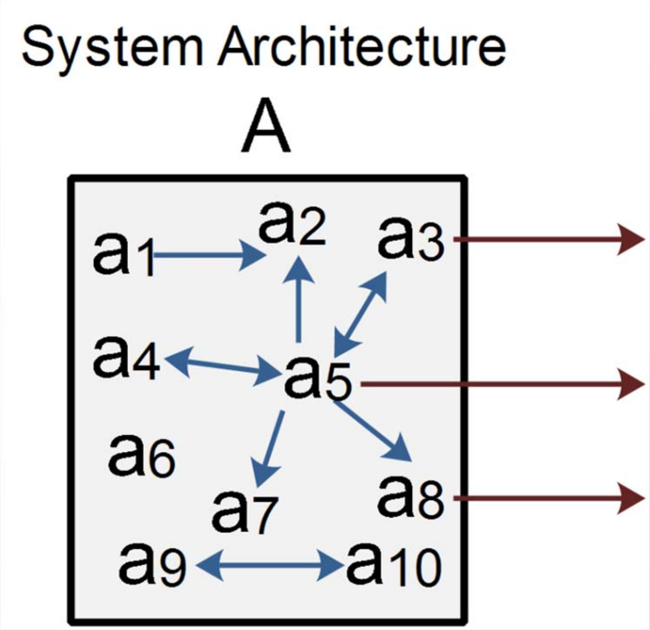
# Specifying and Analyzing a System Architecture

- Many tools exist for specifying a SA
  - e.g.
    - DoDAF 2.0
    - SysML
- Many tools exist for analyzing a SA
  - e.g.
    - Modeling & simulation
    - Functional analysis
    - Use cases
    - N2 diagrams
    - Reliability analysis
- What is necessary is that the system architecture
  - Pliable parameters are clearly identified
  - There is enough detail for the tools to evaluate the instances
    - Subject to decision maker’s satisfaction



# Validating a System Architecture

- Ideally, for a SA to be considered valid, all of its instances should be evaluated and verified to produce an acceptable value under a particular context being considered.
  - An instance does not have to provide value for all contexts
  - All contexts should have at least one instance that provides acceptable value
    - Certain instances may be more important because they are more pliable than others

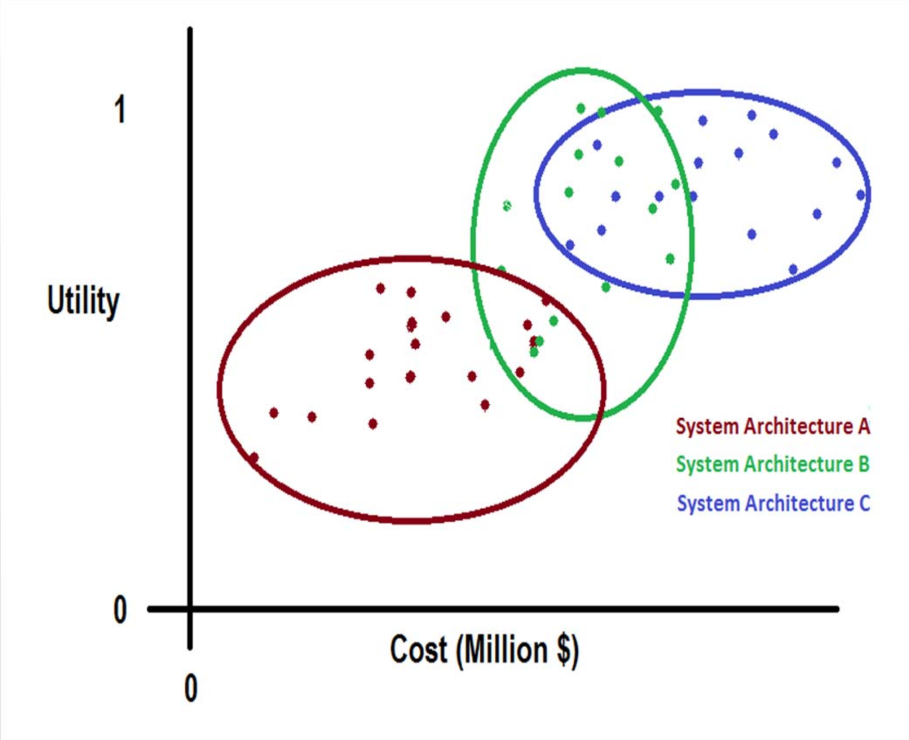



- If the set of possible instances is low and/or evaluations are trivial, then entire SA can be validated
  - Otherwise sampling is needed
  - What designs should be sampled?

# Tradespace Sampling and Pliability

**Setting the pliable ranges provides a good method for tradespace sampling**

1. Start with a specific context and fix all parameters.
2. If value is acceptable, consider other contexts and test.
3. If value is no longer acceptable, increase pliability of SA and test at the limits.
4. Continue to validate new contexts and increase pliability, testing at the limits, as necessary.
5. Eventually, a pliability maximum will likely occur. At this point, change to a new SA and restart process.



# Conclusions

- Evaluating instances of system architecture was proposed instead of point designs
  - CONOPs explicitly included
- Pliability was defined
  - Pliability allows system architects and designers to specify a "warranty" for the system, which defines what is allowed and what is not allowed to change.
  - Pliability may provide latent value since they may make systems more survivable and robust
  - Pliability helps select system designs (instances) to sample for evaluation

# New Research Questions

- Is sampling at the limits of the pliable ranges enough?
  - If not, how should sampling be done?
- What is the best way to represent SA and pliability?
  - SysML, UML, DoDAF 2.0, …
- Are there metrics for pliability?
  - Count the number of pliable parameters, measure the pliable range?
  - What about qualitative parameters?
  - Filtered outdegree?
- How do we compare two SAs without explicitly evaluating instances?
  - Is more pliability better?
- What are design principles for pliability?
- How does pliability fit within other "–ilities"
  - Adaptability, flexibility, evolvability, modifiability, etc.

# Next Steps

- Develop SA for test system
  - Maritime Security Scenario with UAVs
- Evaluate instances of the system architectures using discrete event simulation
  - Expand pliability range and sample at limits
  - Generate new SAs as appropriate
- Examine tradespace to determine correlations between pliability, performance, cost and SA
- Introduce disturbances
  - Re-evaluate existing systems and allow transitions
  - Re-examine tradespace for correlations
- Address new research questions

Questions?