Evolution of Emergent Architectures

NDIA 15th Annual Systems Engineering Conference
25 October 2012

Alan Brenner
(858) 354-2000
al.brenner@bbi2.com

Claudia Rose
(619) 997-5492
claudia.rose@bbi2.com

BBII
PO Box 90182
San Diego CA 92169

Challenges of Today's Defense Programs

- Changing missions
  - Cold War vs Low Intensity Conflicts
  - Dynamic threats
    - IEDs
    - cyberwarfare
- Programs are part of a larger System of Systems
  - Interoperability
  - Integration
- Rapidly evolving technology
  - Technology may be a generation old in 6 months
  - Technology may be obsolete within two years
- Defense programs often have a life cycle measured in decades
  - Field support through end of life can be challenging
  - New systems must be interoperable with legacy and future systems
- Technology refresh of deployed systems elusive
  - Increased performance
  - Lower cost
  - Scalability

What are Complex Adaptive Systems (CAS)?

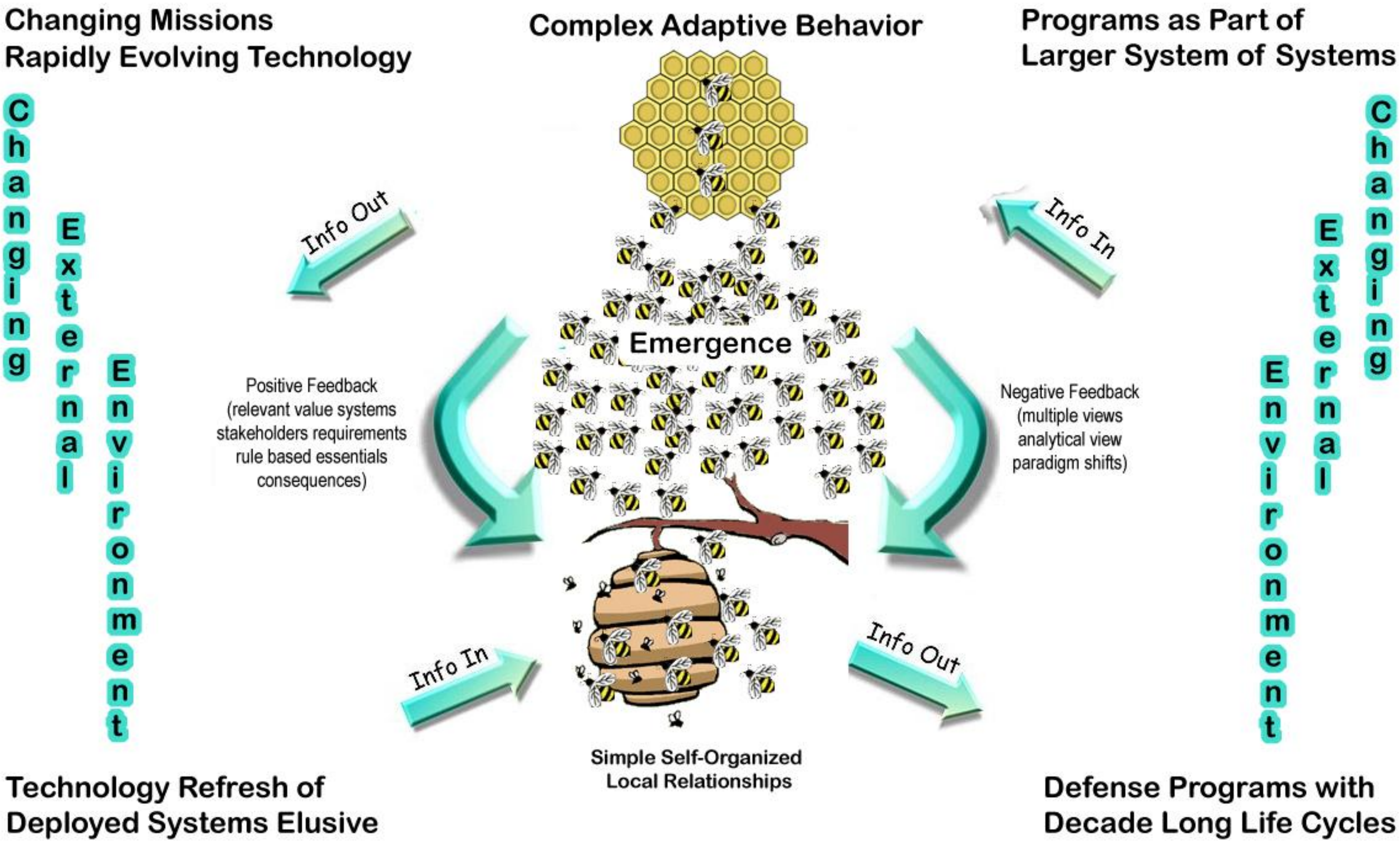
- Dynamic networks of interactions and relationships
- Change their individual and collective behaviors over time
  - Adapting to changing needs
  - Adapting to changing external environment
  - Adapting to changing technology (for synthetic CAS)
  - Adapting to changing manufacturing technology (for synthetic CAS)
  - Adapting to a changing Value System
- Open Systems precluding or obfuscating the definition of System Boundaries
- Elements in the system are ignorant of the behavior of the system as a whole responding only to what is available to it locally
- Control is highly dispersed and decentralized
- Adaption is driven internally for living and intelligent systems
- Adaption is driven by elemental design updates for synthetic systems
- CAS is more than just a Complicated System

Why are Complex Adaptive Systems Needed?

- Lifespan of systems exceeds that of the underlying components
  - DoD and Civil government programs have a lifecycle measured in decades
  - COTS items have a lifecycle measured in months or years
  - Moore's Law has held true for half a century enabling rapid technical evolution
- Current systems are costly to maintain and have outdated capabilities
- Multiple paradigm shifts occur within the lifespan of major systems
- Systems are too complex to be fully understood by a single person
  - Multiple views are required to support different domains
  - Consistency must be ensured across these multiple views
  - Documentation must provide separate manageable analytical views with supporting processes to ensure integrated composite activity
- System development is collaborative across multiple entities with global and temporal disparity
  - System must be partitioned to allow collaborative development across time and space
- Total system upgrade no longer possible because it is a collection of many systems each with a different timetable

What is Emergence?

- Complex Adaptive Systems arise out of relatively simple interactions and rules
- Properties arise from self-organization (e.g. swarming)
- Weak Emergence: result of interactions at the elemental level
- Strong Emergence: qualities not directly traceable to the system's components
  - Irreducible to systems constituent parts
  - The whole is greater than the sum of its parts
- Not deterministic or pre-meditated
- Evolutionary based on adaptation
- Law of Requisite Variety *[An Introduction to Cybernetics, Ashby (1956) p 124]*
  - *Variety absorbs variety [Beer (1979)]*
  - *Cultivation of variety required to overcome disturbances*
  - *Must achieve dynamic stability without incessant fluctuations (noise) [Shannon (1948)]*
- **The object of the game is to go on playing it.** *[John von Neumann (1958)]*
  - *A system with continuing relevance and value*


Changing Missions
Rapidly Evolving Technology
Complex Adaptive Behavior
Programs as Part of
Larger System of Systems
Changing External Environment
Info Out
Info In
Emergence
Positive Feedback
(relevant value systems
stakeholders requirements
rule based essentials
consequences)
Negative Feedback
(multiple views
analytical view
paradigm shifts)
Info In
Info Out
Simple Self-Organized
Local Relationships
Changing External Environment
Technology Refresh of
Deployed Systems Elusive
Defense Programs with
Decade Long Life Cycles

Examples of Emergent Systems

Weather

GPS Navigation System

Ecosystems

Natural Nuclear Reactor

<u>Societies</u>

Bee Colony

United States of America

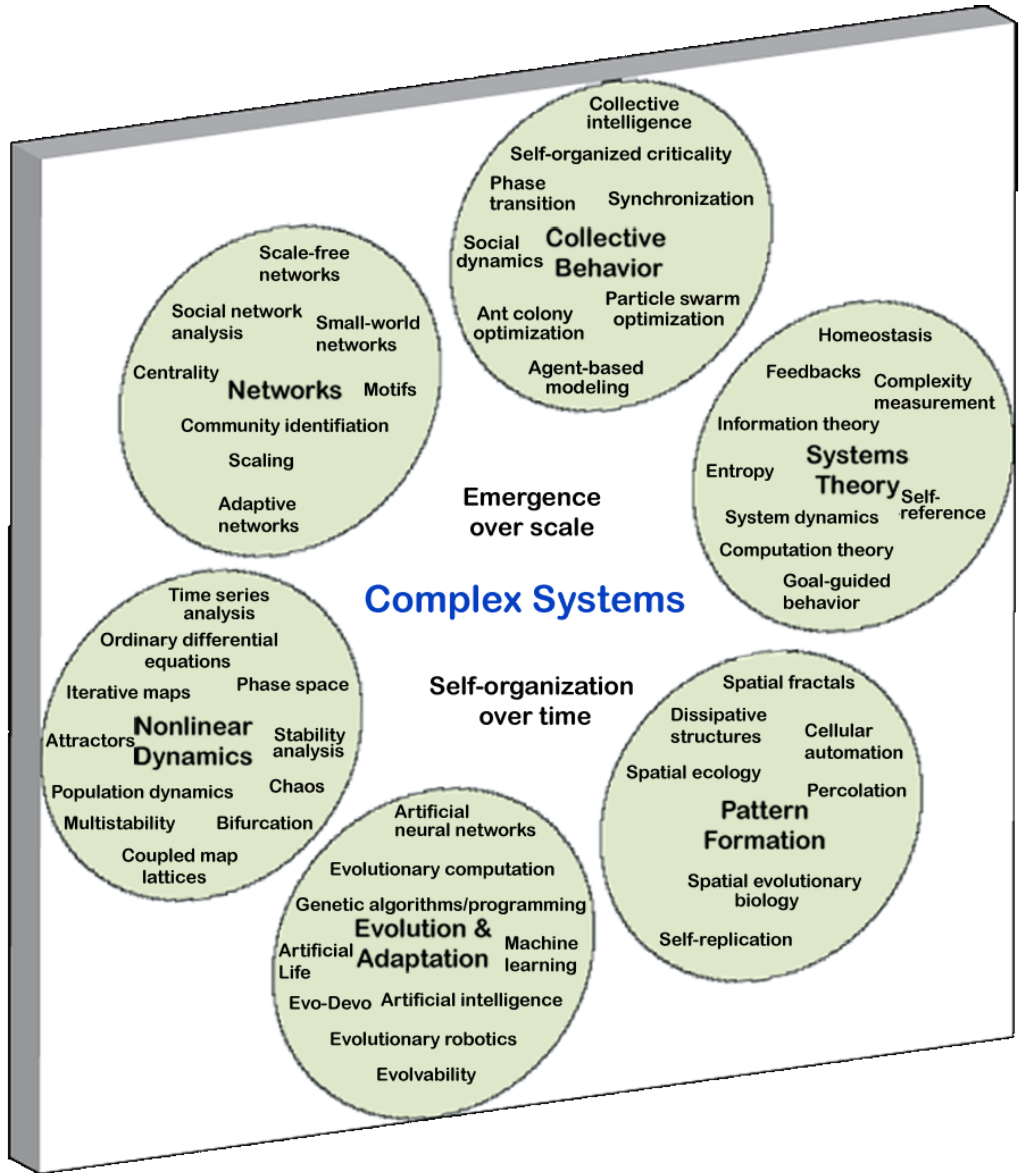
<u>Synthetic</u>

Interstate Highway System

Cities

Internet

Social Networking Service



Emergence of Natural Nuclear Reactor at Oklo Uranium Mine in the Gabon Republic, West Africa

- Emerged from bacteria in an algal mat concentrating uranium *(at that time the uranium would not have needed enrichment)*
- Bacteria thrived in the radioactive environment
- Ran for a million years with power output on the order of 100 kilowatts about 2 billion years ago
- Discovered in 1972 when the uranium ore was 200 kg short of U^{235} because it was consumed in the reactor
- Caused considerable concern as to where the U^{235} was diverted before investigations lead to this discovery

Example of GPS Navigation

- Goal Oriented
- Rule based
- Adaptive to real-time updates as available (e.g. traffic conditions)
- Adaptive to driver taking a different route or error
- Will always provide some solution if a possibility
  - e.g. swimming from pier in New York to France
  - Many units will show jeep roads in local area if off paved roads
- Market of products provides requisite variety
- Consumer choices will result in emergence of new capabilities
- Smart cars will network with others resulting in emergent network dynamics

GPS Navigation Like Situational Awareness

- Examples of gateway target exploits
  - Root account
  - Local Security Authority (LSA)
  - Trusted Root/Intermediate Certificate Store
  - Routing Table Control
  - Active Directory Group Policy vulnerabilities
- Exploit Path (may be chain of exploits)
  - Open / Open Without Detection / Open with Zero Day exploit
  - Solution indicates vulnerability/exploit – lack of solution
- Ultimate goal is a picture of red/blue operational capabilities
  - **What can I have?**
  - **What can they get?**
  - Includes probabilities of success and risk aggregation
  - Identifies resources needed for mission success
- System needs to operate in real-time with emergent and adaptive behaviors
- Cloud enables essentially unlimited computational resources as needed

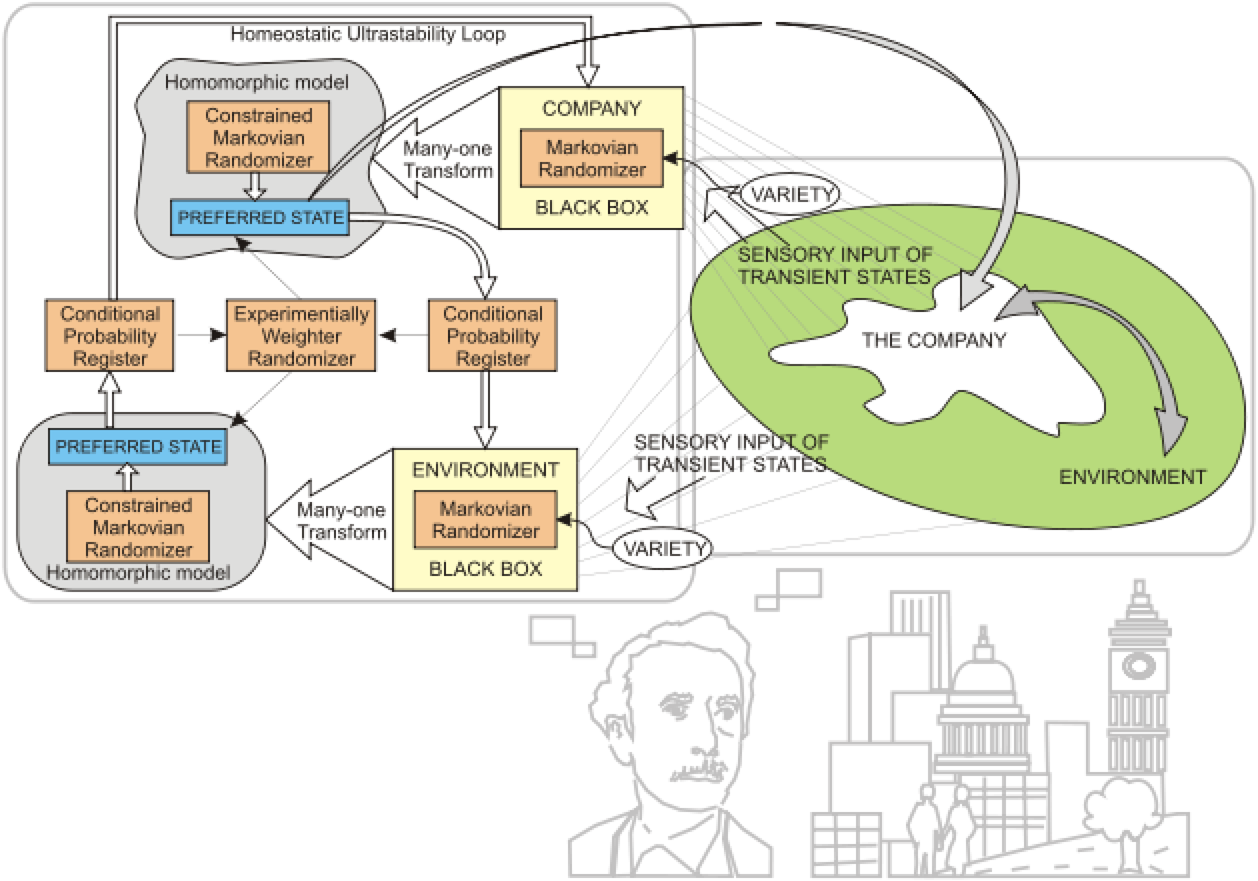
Biological Threat Response is Distributed

- Sustaining defense mechanisms always in place (e.g. white blood cells, antibodies)
  - Threats are engaged autonomously (no decision from brain required)
- Local response such as increased circulation, swelling, pain contribute to defense and protection
- Markers in blood stream will signal for elevated response distributed through general circulatory and lymphatic system
- There is typically no involvement of brain to require deployment of defenses
- Awareness of threats through sensory system and general feelings of health allow for actions such as rest to promote healing
- Mechanisms also operate at level of populations for collective benefit of society
  - Recent hyper-mobility of organisms & people has substantial consequences
  - Entire ecosystems changed
  - Evolutionary timescale greatly compressed

What are Challenges of Developing Complex Adaptive Systems?

- Defining the problem
  - "We have also come to realize that no problem ever exists in complete isolation. Every problem interacts with other problems and is therefore part of a set of interrelated problems, a system of problems. ...I choose to call such a system a ***mess***." *[Ackoff (1974)]*
  - Mess is changing over time
- Defining a system which is constantly evolving
  - Current modeling tools are a snapshot and do not reflect evolving system
  - Mainstream best practices emphasize repeatability, discourage variety and stifle innovation
  - Identifying deep patterns which foster emergence
- Acceleration of emergence
  - Incubate variants which will thrive
  - Curtail resources expended on variants which are suboptimal
  - Avoid the perils and risks of eugenics and unintended consequences
- Establish a relevant value system which is not corrupted by the observations
- Create the incubator from which the system emerges
- Understand the consequences of each decision
  - "Progress imposes not only new possibilities for the future but new restrictions." *Norbert Weiner (1950)*

Not only must we build the system, but we must build the incubator (factory) for the system



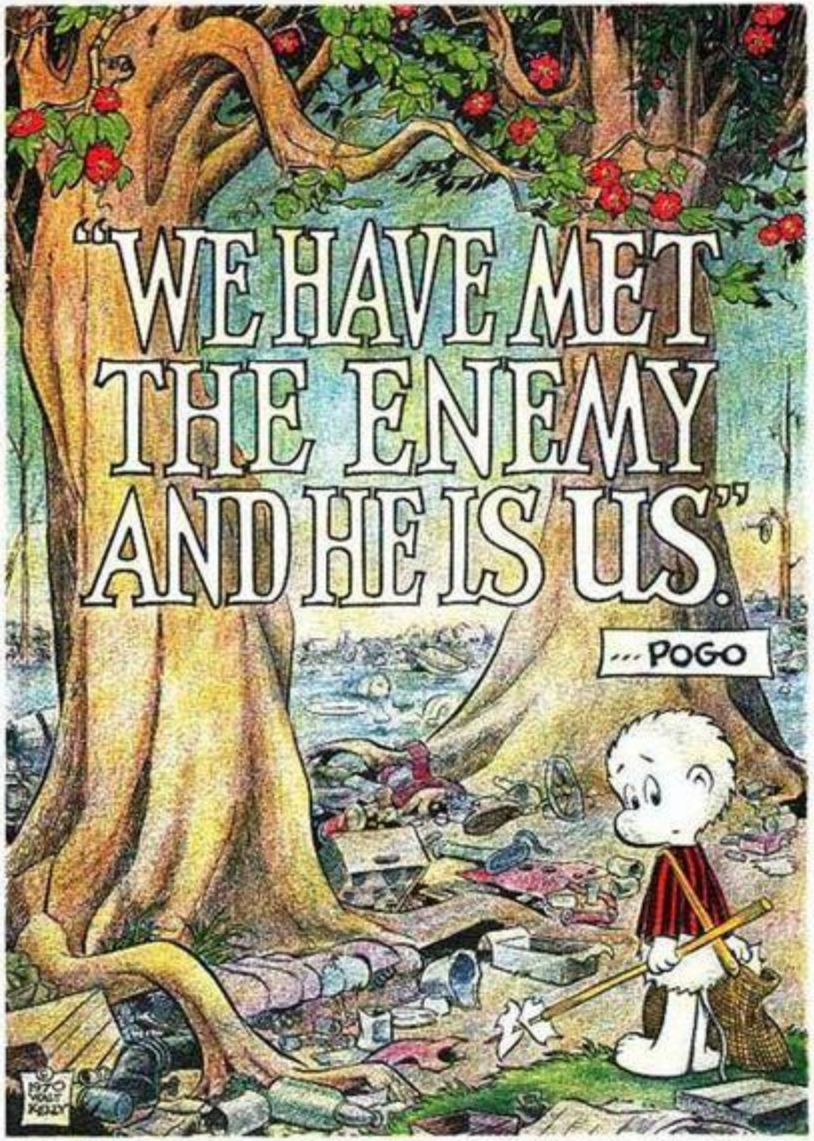
http://en.wikipedia.org/wiki/File:Model_Cybernetic_Factory.svg

Zero Time is Built Into the Enterprise Architecture with Emergent Qualities

- Instant Adaption
  - Requisite variety part of product structure to maximize options for adaption
  - Goal oriented architecture continuously adapts
- Instant Involvement
  - All actors of enterprise are connected through Information Repository driven by the relationships of the Information Model
- Instant Value Alignment
  - Value system integrated with Information Model
  - Value system has all the emergent qualities and drives the enterprise solution
- Instant Execution
  - Requisite Variety, emergent value system, and enterprise wide integrated information model all contribute to the capability for Instant Execution
- Instant Learning
  - Information Workflow is integrates within the repository, which is continually expanding
  - Domain Knowledge is a key part of the Information Model along with Decision Rationale
- All five qualities achieve Perpetual Market Lock
  - Any four achieve Market Lock
  - Any three are a Market Leader

Classical Systems Engineering is No Longer Sufficient for the Solution

- *Albert Einstein* observed "We cannot solve our problems with the same thinking we used when we created them."
- CMMI has brought us processes which ensure repeatability across programs
  - The systems still come in over budget, behind schedule and fall behind technologically
  - We then strive for the next level of CMMI
  - Insanity: doing the same thing over and over again and expecting different results. *Albert Einstein*
- Any intelligent fool can make things bigger and more complex... It takes a touch of genius - and a lot of courage to move in the opposite direction. *Albert Einstein*
  - Waterfall, V-Model, Spiral Development expand the scope (and complexity) in Classical Systems



Walt Kelly's poster for the first Earth Day

http://en.wikipedia.org/wiki/File:Kellyposter1970.jpg

Classic Decomposition and Specification may have Unintended Consequences

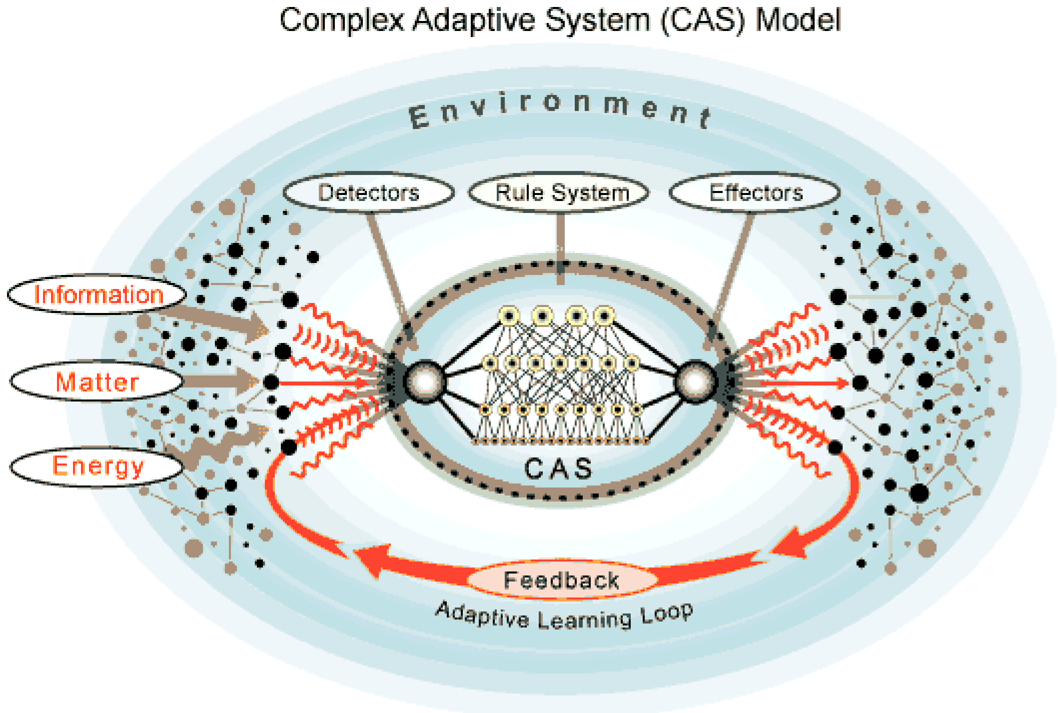
- From Christian Science Monitor article regarding the RQ-170 Sentinel
  - Forced to return home by loss of command link
    - Link jammed
    - Default procedure
  - Spoofed GPS data to think location in Iran was home Landing Zone
  - LZ was off a few feet in elevation resulting in minimal damage upon landing
  - Report certainly raises many questions – however example is only illustrative
- Ariane 5 First Launch (June 4, 1996)
  - Primary Inertial Reference System (SRI) switched offline due to software error
  - Secondary SRI was already offline due to same error on hot standby
  - Resulted in full nozzle and destruction at about 39 seconds into flight
- Illustrates emergent behaviors are desirable to better achieve mission goals
  - Goal oriented behavior to achieve mission success
  - Need for requisite variety
  - Autonomous actors to limit negative system influences
    - independent verification
    - independent corrective action

What is an Emergent Architecture?

[Gartner – 2009]

- Non-deterministic: *In the past, enterprise architects applied centralized decision-making to design outcomes. Using emergent architecture, they instead must decentralize decision-making to enable innovation.*
- Autonomous actors: *Enterprise architects can no longer control all aspects of architecture as they once did. They must now recognize the broader business ecosystem and devolve control to constituents.*
- Rule-bound actors: *Where in the past enterprise architects provided detailed design specifications for all aspects of the EA, they must now define a minimal set of rules and enable choice.*
- Goal-oriented actors: *Previously, the only goals that mattered were the corporate goals but this has now shifted to each constituent acting in their own best interests.*
- Local Influences: *Actors are influenced by local interactions and limited information. Feedback within their sphere of communication alters the behavior of individuals. No individual actor has data about all of an emergent system. EA must increasingly coordinate.*
- Dynamic or Adaptive Systems: *The system (the individual actors as well as the environment) changes over time. EA must design emergent systems sense and respond to changes in their environment.*
- Resource-Constrained Environment: *An environment of abundance does not enable emergence; rather, the scarcity of resources drives emergence.*

The Question remains: How does one actually build Emergent Complex Adaptive Systems?

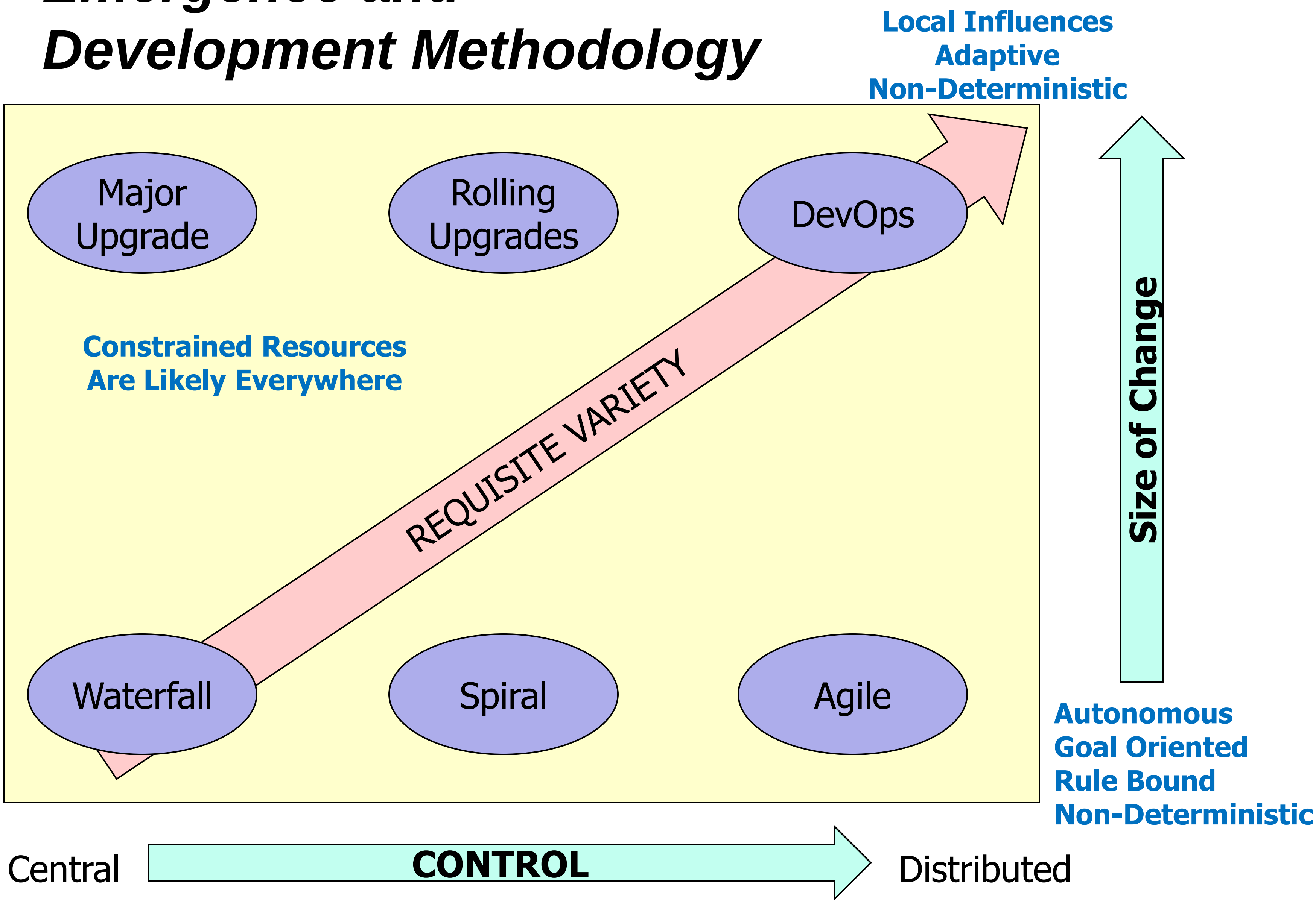


http://www.calresco.org/lucas/cas.htm

Development Methodologies

- No single methodology is best for all projects
- Methodology should be chosen based upon desired goals of enterprise architecture
- Regions within system layers and boundaries:
  - Can use different methodologies
  - Operate on different time scales
  - May use different enterprise architectures
  - Scale ranges from subatomic solid state devices to global systems
- e.g. DevOps at the application level, but not the core infrastructure level
  - DevOps merges agile, IT operations, and quality methodologies
- Need for a comprehensive end-to-end is often argued
  - Works pretty well to leave processor manufacturers alone
  - Market drives technical evolution
  - Works because boundaries are strong and interfaces are well understood

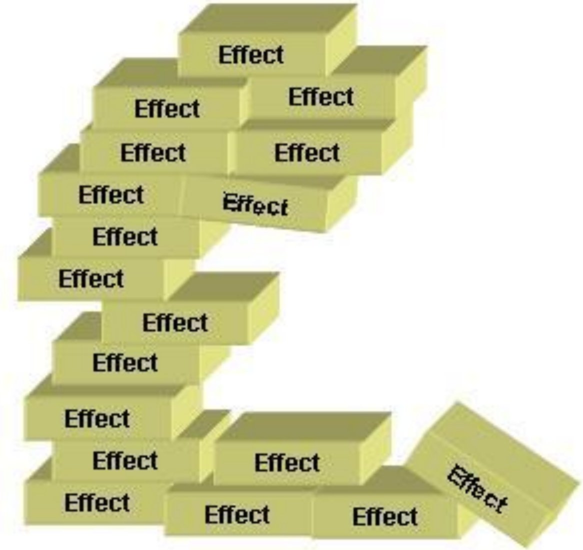
Emergence and Development Methodology



Architecting for Emergence

- Focus on the boundaries, interfaces, rules and interactions of the entities
- Identify and characterize the deep structures
- Minimize specifications of behavior which impede evolution and innovation
- Characterize domains of Emergent Complex Adaptive Systems
  - The Mess
  - The Environment
  - The Enterprise
  - The System
- Tie the solution back to the problems using the value system to evaluate variants
  - Continuing emergence through lifecycle of program
  - Feedback through value system defines emergent properties
- Create Information Model and automate information flow
  - Create views to support task at hand
  - Maintain referential integrity of data through automation
  - Too much information is available to effectively use unfiltered
  - Facilitate effectivity and impact assessments
- Distribute and delegate decision authority to lowest levels possible

The Mess



http://www.fs.fed.us/psw/topics/fire_science/craft/craft/Resources/assets/CumulEffects_blocks.jpg

- Approach as a system of interdependent problems
- Identify stakeholders, requirements and values
  - Contradictory
  - Incomplete
  - Changing
  - Define in native language
    - Stakeholders do not think in context of foreign models
    - Translate if necessary to Information Model
- Identify entities, roles and interactions
  - Characterize through models
  - Characterize evolution over time
- Identify constraints
  - Resources
  - Political
- Define Value System

WICKED PROBLEM

http://robertweber.typepad.com/photos/uncategorized/2007/09/22/cartoon2_4.jpg

The Environment

- Characterize environment through models, behaviors and constraints
  - Layered models organize decompositions
  - Create multiple views representing each domain
  - Identify boundaries, interfaces and deep structures
  - Characterize evolution over time
- Characterize other systems which are part of the environment
- Characterize legacy systems which are both constraints and opportunities to leverage existing infrastructure
- Understand the System Dynamics before making changes
  - Deep structures provide opportunities for emergence
  - Minimize negative unintended consequences (outcome could also be positive)
- Characterize emerging and existing technology
  - Maturity
  - Interfaces
  - Capabilities

The Enterprise

- Approach as Enterprise of Enterprises
- Define Information Model
  - Allow for temporal evolution
  - Allow for element diversity
- Define Operational Behavior Models of Enterprise
  - Governance
  - Workflow
- Define System Models
- Embed workflow processes in model infrastructure
- Implement and deploy models in tools and databases
- Extend modeling tools for temporal diversity
  - Current generation tools support snapshots at an instance in time
  - Changing relationships of an entity over time is not handled well in current tools
- The Enterprise which builds the system is a Complex Adaptive System in itself
- Emergent Enterprises are distributed in nature and cannot be centrally managed to the lowest level

The System

- Architect Emergent Complex Adaptive System
- Commit to tool based infrastructure using Information Model
  - Only way to ensure referential integrity of complex system
  - Workflow is embedded in tool infrastructure
  - Reviews of views ensure composite integrity
  - Use information model to assess effectivity of system
  - Use information model for impact assessment of architectural changes
- Classical Systems Engineering techniques are counter-productive
- Architecture is layered with corresponding models
- Focus on boundaries and interfaces between elements within layer
  - Respect boundaries between layers
- Innovate with novel approaches to meet the operational needs
  - Question authority – tradition is not a constraint
- Procured solution instances are different than architectural components
  - May be fully specified to allow life cycle support
  - May be replaced with new instances with enhanced capabilities fitting into the existing architecture

Summary

- "The object of the game is to go on playing it." *John von Neumann (1958)*
- "Progress imposes not only new possibilities for the future but new restrictions." *Norbert Weiner (1950)*
  - Every specification (requirement) is also a constraint
- "We cannot solve our problems with the same thinking we used when we created them." *Albert Einstein*
  - Repeatability is not necessarily your friend
  - "Insanity: doing the same thing over and over again and expecting different results." *Albert Einstein*
  - Law of Requisite Variety *Ashby (1956)*
- Must Architect for emergence – it will not happen by itself in a suitable time frame
  - Need Enterprise Architecture AND System Architecture
  - Minimal specifications
    - Any intelligent fool can make things bigger and more complex... It takes a touch of genius - and a lot of courage to move in the opposite direction. *Albert Einstein*
    - Every specification (requirement) is also a constraint
- Focus on objectives – specify what needs to happen, not how
- Outcome is non-deterministic – it is rule bound and goal oriented
  - "It's extremely difficult to second guess the American Navy, because the Americans rarely read their doctrine, and don't feel compelled to follow it when they do."
    Attributed to Admiral of the Soviet Fleet, Sergei Gorshkov