Honourcode, Inc.

Complexity in Systems of Systems

An introduction to the theoretical basis for SoS

- *Systems Engineering*
- *Training Courses*
- *Process Improvement*

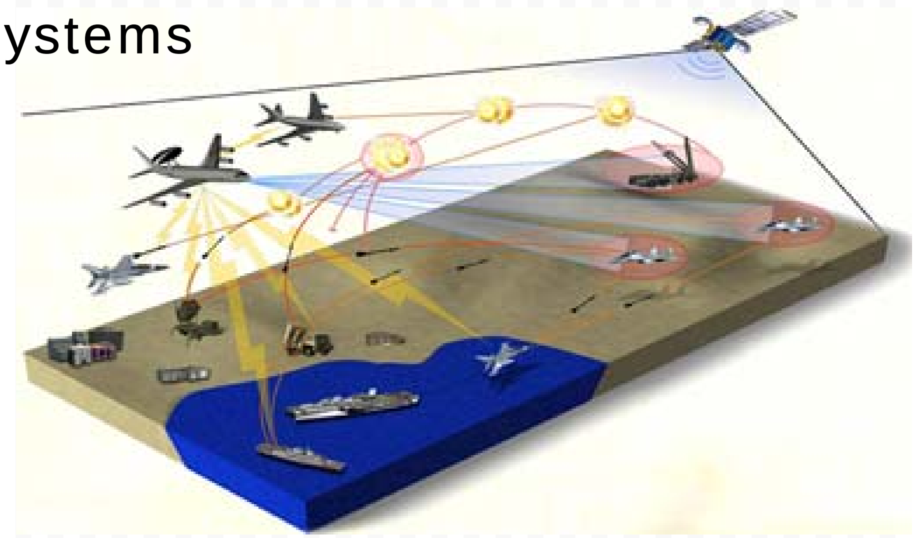
Scott Workinger, Ph.D.
+1 (650) 363-0979
sworkinger@hcode.com

SoS Example: Supply Chain Mgmt

- Component systems
  - Production management systems
  - Inventory systems
  - Transportation tracking systems
  - Internet for connectivity
- Functions
  - Reduce inventory costs
  - Just-in-time inventory production
- Development
  - Each system developed separately
  - Little coordination
  - Systems upgraded separately

SoS Example: a Military Force

- Component systems
  - Aircraft, ground vehicles, ships, submarines
  - Each with unique sensors, communications, weapons
  - Soldiers with instrumented personal equipment
  - Many communications systems
  - Command/control (C^2)
- Functions
  - Force projection
  - Battle superiority
  - Command/control



- Development
  - Vehicles, C^2, equipment, sensors, comms, weapons
  - Each system developed separately
  - Developments occurred over decades of time

SoS Example: a Modern Airport

- Component systems
  - Aircraft, support/maintenance, baggage handling
  - Air traffic control, ground control, taxiways, runways
  - Ticketing, reservations, gate control, boarding bridges
  - Transportation security
  - Parking, auto traffic control
  - Shopping, pedestrian control
- Functions
  - Passenger transportation
  - Flow control
  - “Enjoyable experience”

- Development
  - Some systems developed together when airport built
  - Others developed separately over decades of time
  - Different customers, different developmental control

Systems of Systems Defined

What’s the difference between a “system” and a “system of systems”?

SoS Characteristics

A System is a "System of Systems" if:

- Operational independence - component systems have purpose even if detached
- Managerial independence - component systems are developed and managed for their own purposes
- Evolutionary development - functions and purposes are added, removed and modified in an ongoing way
- Emergent behavior - SoS performs functions not achievable by the independent component systems
- Geographic distribution - geographic extent forces the elements to exchange information in a remote way

- Mark Maier

Operational Independence

Military Force

- Fighter aircraft
  - Force at a distance
  - Protect own forces
  - Prevent enemy intrusion
- Navy frigate
  - Worldwide mobility
  - Control sea lanes
  - Escort non-combatants
  - Littoral strike
- Mobile howitzer
  - Ordnance delivery
  - Rapid movement

Airport

- Tracking radar
  - Sense aircraft in area
  - Situation awareness
- Ticketing/reservations
  - Flight reservations
  - Passenger control
- Baggage handling
  - Receive baggage
  - Route baggage to aircraft
  - Deliver baggage
- Boarding bridge
  - Adaptable access to different aircraft

Managerial Independence

Military Force

- Services have separate acquisition organizations
  - Army
  - Air Force
  - Navy
  - Marines
- Government budgeting process demands line item breakout
- Each system acquired, maintained independently
- SoS includes captured and enemy systems

Airport

- Systems owned by different agencies
  - Airport authority
  - Separate airlines
  - Transportation security
  - Retail
- Some systems managed conjointly by airport
- Other systems are completely independent

Evolutionary Development

Military Force

- Technology change
  - Add new planes, ships
  - Prevent obsolescence
  - Add new possibilities
- Mission change
  - New threats
  - Different types of operations & enemies
- Operational change
  - New joint operations
  - Operate with new allies

Airport

- Technology change
  - New airline ticketing systems
  - Link rental cars with frequent flyer programs
- Cost savings
  - Implement cost-saving functions
  - Replace aging equipment
- Mission change
  - Mitigate terrorist threats
  - Link airport retailers with ticketing systems

Emergent Behavior

Military Force

- Force projection
  - Provide military presence at a forward location
  - Enforce policy
- Battle superiority
  - Win against enemy forces
  - Overpower opposition
- Command/control
  - Precise control of actions
  - Effective placement

Airport

- Passenger transportation
  - Passenger & baggage paths to/from aircraft
  - Entertainment, revenue
- Flow control
  - Vehicle entrance, parking
  - Ticketing, control
  - Appropriate waiting areas
  - Passenger vs. non-passenger control
  - Controlled access areas

Individual component systems do not provide these behaviors, except when part of the SoS

Geographic Distribution

Military Force

- Ships
  - Cohesive force scattered across 100s miles
- Aircraft
  - Many bases and aircraft carriers
  - Individual missions
- Armies
  - Coordinated operations across an entire theater

Airport

- System locations
  - Airport terminal
  - Aircraft
  - Internetted
- Distribution
  - Communications through Internet, radios
  - Cross-agency information

Impacts of SoS Characteristics

Operational Independence

Component systems change; little insight into their structure

Managerial Independence

Lack of control of the SoS; limited effect of guidance

SoS "System Control" is a significant challenge

Stakeholders want assurance of the emergent behavior

Emergent Behavior

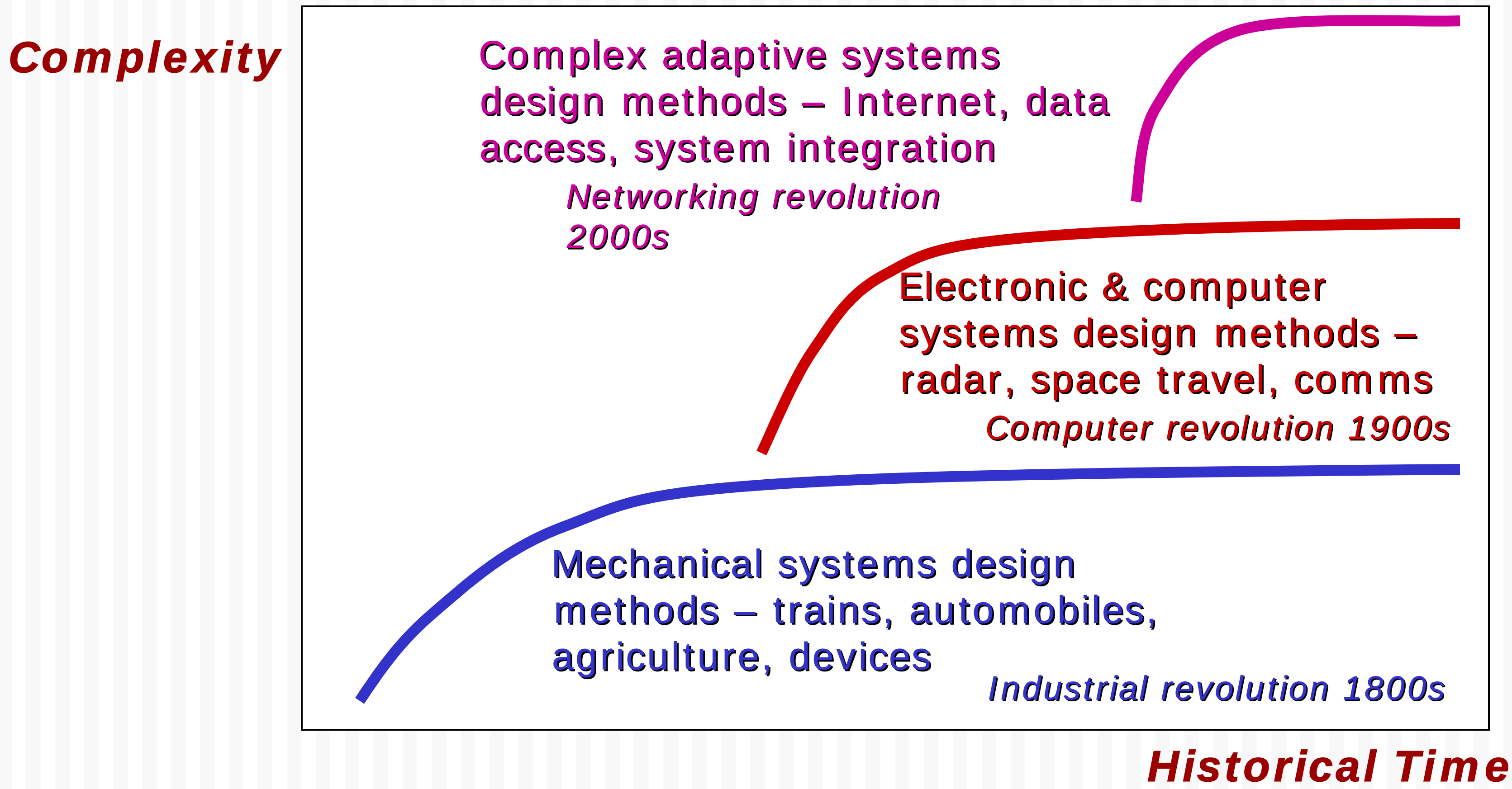
Difficult technical control of timing, interfaces

Geographic Distribution

Constant revision, integration issues

Evolutionary Development

Paradigm Shifts in Engineering



- **Each paradigm fuels a rapid growth and then stagnates as it tries to handle more complex products**

Complexity Issues

What is complexity?

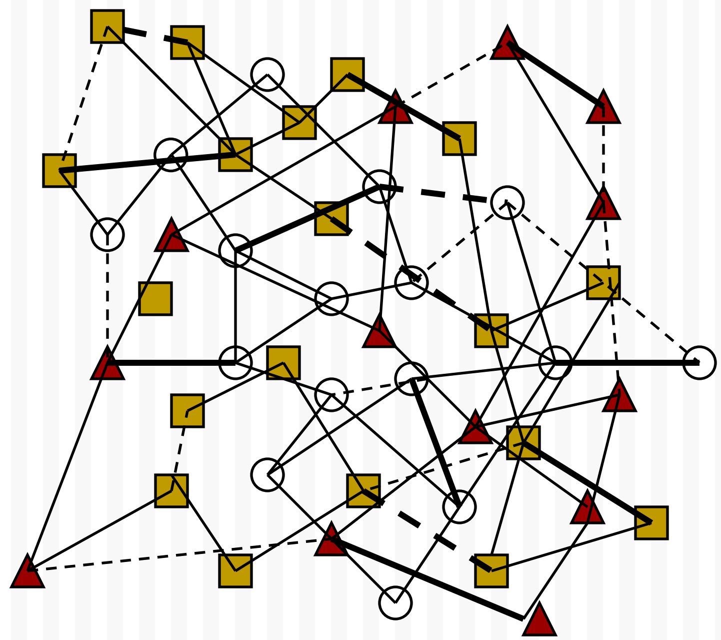
Why does it matter?

Complexity

- ***Having so many interacting components that behavioral phenomena are inexplicable by any conventional analysis.***

Examples

- Stock market
- Weather
- Biology
- Ecosystem
- National economy



- *Inherently subjective*
- *Informal usage*

Complexity on the Edge of Chaos

Deterministic Systems (may be complicated)

Predictable using determinism

Complex Systems Are Dynamic

Chaotic Systems (may be simple)

Predictable using statistics

Delicate balance
Between order and chaos

Spontaneous, Active, Alive!

Enough stability to persist (but what persists is patterns, not components)

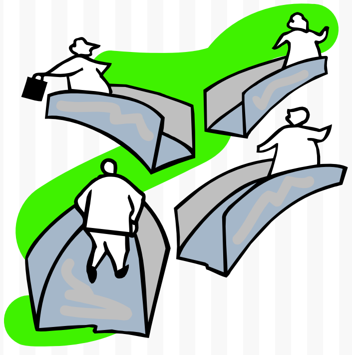
Enough changeability to respond (but what changes is behavior, not composition)

Properties of Complex Systems

Property	*Ordinary Systems*	*Complex Systems*
Predictability	No surprises	Surprise behavior
Connectedness	Few components, Simple interfaces, Little feedback	Many components, Complicated interfaces, Much feedback
Control	Centralized or few, Traceable, Fixed behavior	Diffusion of control, Nontraceable, Adaptable
Decomposability	Weak interactions, Severable components, Decomposable	Many interactions, All elements essential, Irreducible
	Toaster	*Internet*

Some Effects of Complexity

- Small worlds
- Patterns
- Emergent properties
- Self-organization



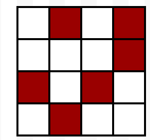
These effects (and others) are normal in complex systems, and they present difficult challenges to engineer.

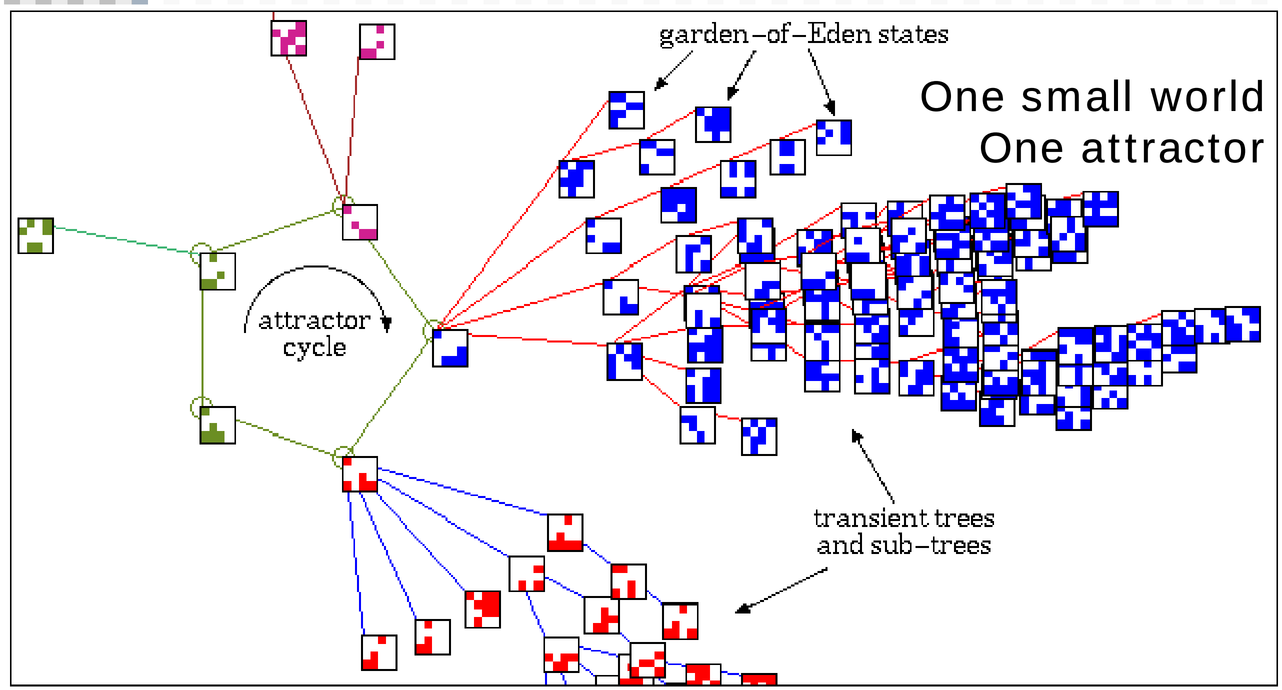
For each of these, we will show what it is, an SoS example, and design methods usually attempted

Small Worlds

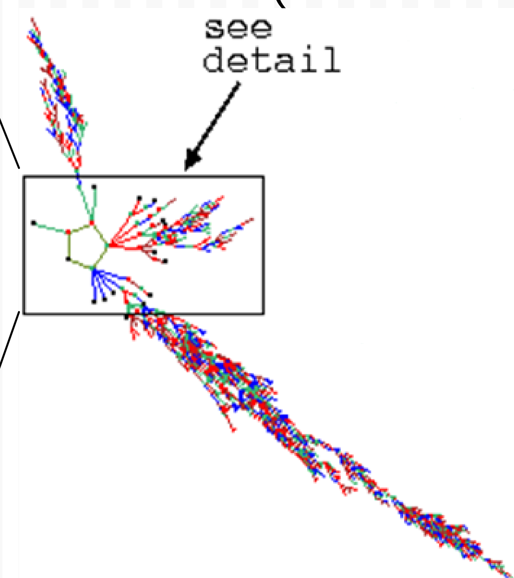
- First observed in cellular automata
- Small set of rigid rules → complexity!
- 64K states organize into 11 small worlds
- Each small world has an attractor cycle

Cellular automaton 4x4 cells rigid rules



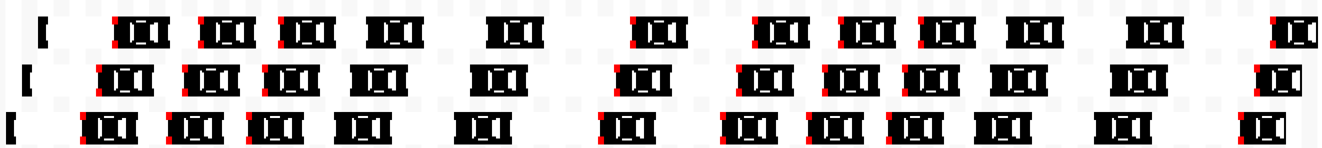


One small world state map (512 states)



Small Worlds in an SoS

- Traffic control in a metropolitan area
  - Daily flow of traffic in, out, around
  - Many control systems, independent and/or linked
- Result is many small worlds, each with its own attractor states
  - Neighborhoods with traffic centers
  - Flow routes with usual jam sections
- Perturb the flow and…
  - It diverges to new states
  - Then returns to attractors



Working with…
Small Worlds and Attractors

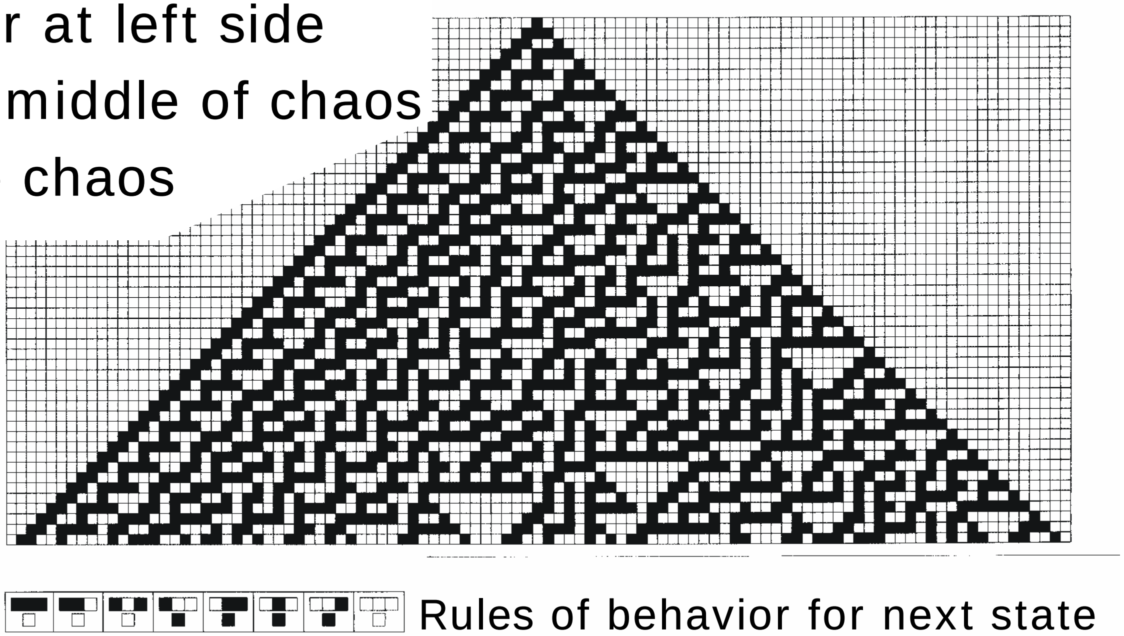
- Simulation and analysis
  - Identify the groupings that are the small worlds
  - Separate the analyses
- Architecting
  - Identify the attractor states
  - Architect to make them desirable
  - Result is SoS robustness

Patterns

- Also seen in cellular automata – example shows successive states of a one-row automaton
  - Simple rules, nearly symmetric
  - Note the complexity of the result
- See the patterns
  - Regular behavior at left side
  - Triangles in the middle of chaos
  - Fishhooks in the chaos

Humans see the patterns

Systems only react to them

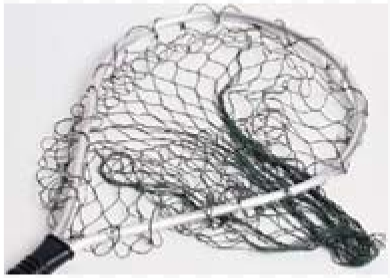


Rules of behavior for next state

Patterns in an SoS

- Communications patterns in a military force
  - Even with networked <u>structure</u>, comms <u>patterns</u> vary
- Hub-and-spoke pattern
  - Central command authority with high traffic density
  - Subordinate units with much lower density
  - Pattern name: wheels

- Networked pattern
  - Co-equal units providing information
  - Each unit draws its information as needed
  - Pattern name: fishing



Pattern depends on the individual agents – difficult to force any pattern to happen

Working with…
Patterns

- System Architecting
  - Use patterns as heuristics for architectural design
  - Gain benefits of each pattern used
  - Develop a career-long memory of successful patterns
- System Development
  - Use development patterns also
  - Process definitions are known patterns
  - Use “lessons learned” from others
- Problem Analysis
  - Recognize patterns in the SoS
  - Use patterns to seek causes, solutions

Emergent Properties

- **Emergence** definition
  - Complex pattern formation from more basic constituent parts or behaviors
  - "Properties of the whole that are not evident from the parts."
- Emergence happens when
  - Parts perform their own functions
  - Relationships among the parts influence how the parts operate
  - Influence is local in nature
- Emergence often not predictable from the nature of the parts.

Termite cathedral mound

Emergence in an SoS

Some emergent properties of the airport

- Flow of passengers
  - Parking → ticket counter → security → gate → airplane
  - Airplane → connecting gate → airplane
  - Airplane → baggage → rental car → exit
- Security control
  - Open zone / controlled zone / airplane
- Airplane control
  - Air traffic radars, communications
  - Ground traffic observation, communications
  - Gate assignment, routing, baggage, passengers
- Crowd solicitation by independents
  - Crowding of passengers draws opportunists

Types of Emergent Properties

Emergent properties are an important part of all systems engineering!

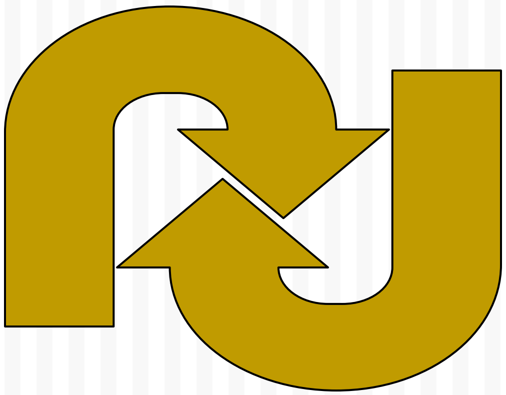
Some emergent properties of an automobile.

	Useful	Neutral	Destructive
Designed	Desired capabilities *transportation*	Facts of design *physical size*	Accepted trade-offs *braking distance*
Surprise	Exploitable features *quiet place*	Facts of existence *heat generator*	Fearful features *killing potential*

Working with…

Emergent Properties

- Top-down design
  - Identify the desired emergent properties
  - Architect the SoS – component systems and their relationships – to create the desired emergence
  - Identify the acceptable trade-offs
- Bottoms-up integration
  - Simulate and test to find the emergent properties
  - Check for presence of the desired properties
  - Evaluate the acceptable trade-offs
  - Discover the surprise properties



Self-Organization

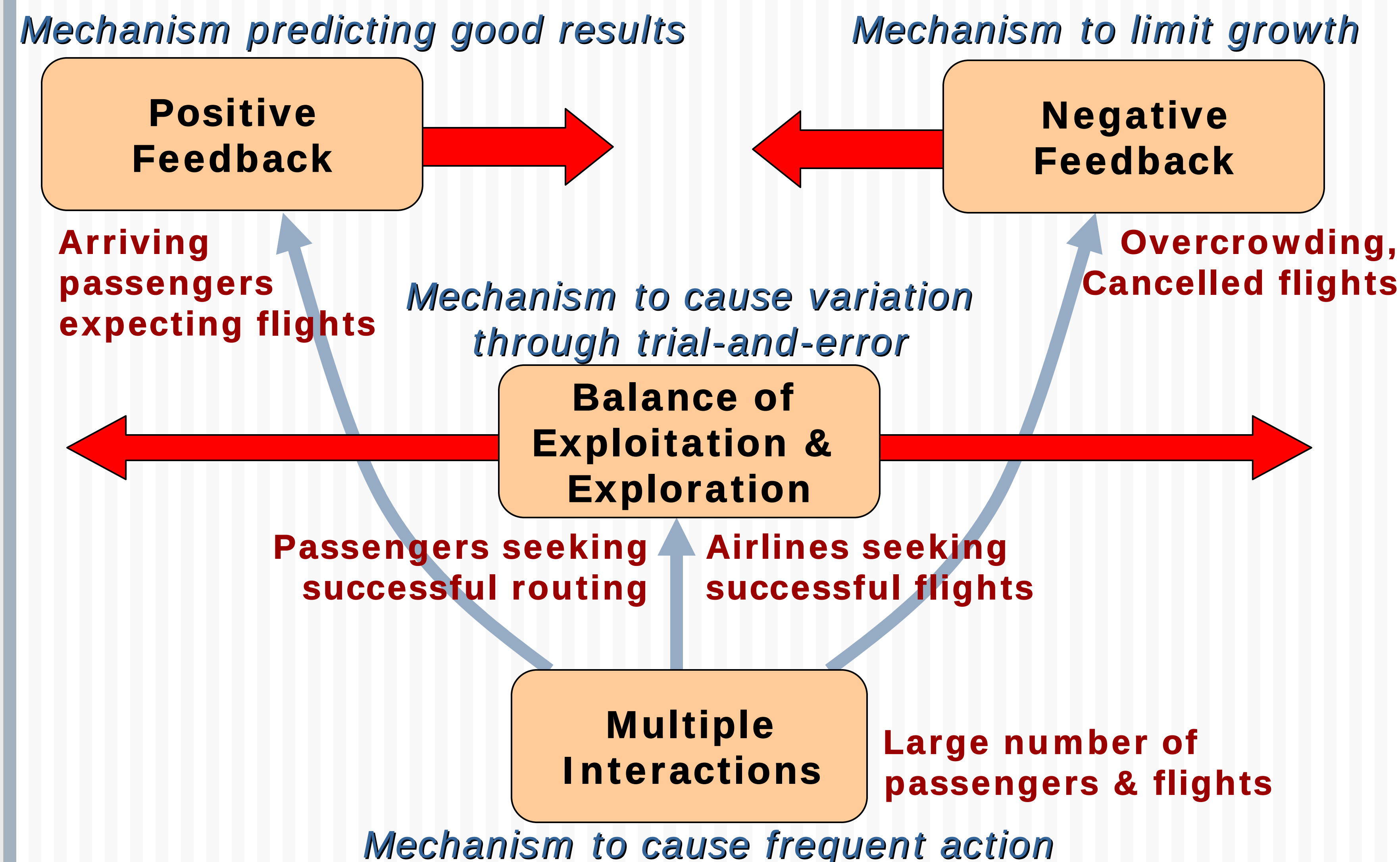
- **Self-Organization** definition
  - Internal organization of a system increases in complexity without being guided by an outside source
  - Often coupled with emergent properties
- Examples exist in physics, chemistry, biology
  - Propagation of a rip in a sheet of paper
  - Turbulence vs. laminar fluid flow
  - Self-assembly of enzymes
  - Autocatalysis, a cycle of catalysts
  - Morphogenesis, how an embryo grows
  - Flocking of birds, fish, insects

Self-organizing smoke

Self-Organization in SoS

- In a military force during operation:
  - Units cooperate ad hoc to storm a position
  - Soldiers discover that a sensing laser causes enemy response – start using the laser in new ways
  - Situation tracking system becomes a mapping system
- In an airport:
  - Take-off order of flights is changed by a delay in the automated baggage handling system
  - Passengers congregate early to board at a gate; other flights are delayed by the crowding

Causes of Self-Organization



Working with…

Self-Organization

- Recognize the opportunity
  - SoS can improve over time
  - New ideas, new relationships
  - Growth and change
- Embrace and use the self-organization
  - Create robust structures of positive, negative feedback
  - Implement goals visible to the SoS
    - Cause exploitation, exploration, goal-seeking
    - Goals may be human or machine automated
  - Monitor and guide through the goals, structures

Complexity Summary

The bad news:

- Systems of systems have sufficient scale to cause complex behavior
- As a result, in the world of SoS, not everything is controllable or planned

The good news:

- Complexity theorists have been working on understanding this behavior for many years
- SEs are beginning to turn the theory into practice

A Solutions Framework

Architecture Solutions
- Capability engineering
- Architecture frameworks
- Self-organization
- Patterns

Collaboration Solutions
- Working w/ multiple teams
- Concurrent SE
- Program interfaces
- Collaboration tools

Integration Solutions
- Interface control
- Coupling & interoperability
- Open systems
- COTS integration
- Legacy system integration

Test & Eval Solutions
- Multiple levels of T&E
- Evaluating interfaces
- Validating functions
- Evaluating dynamics
- Evaluating emergence

Systems engineering is the engineering of complexity!

More Information?

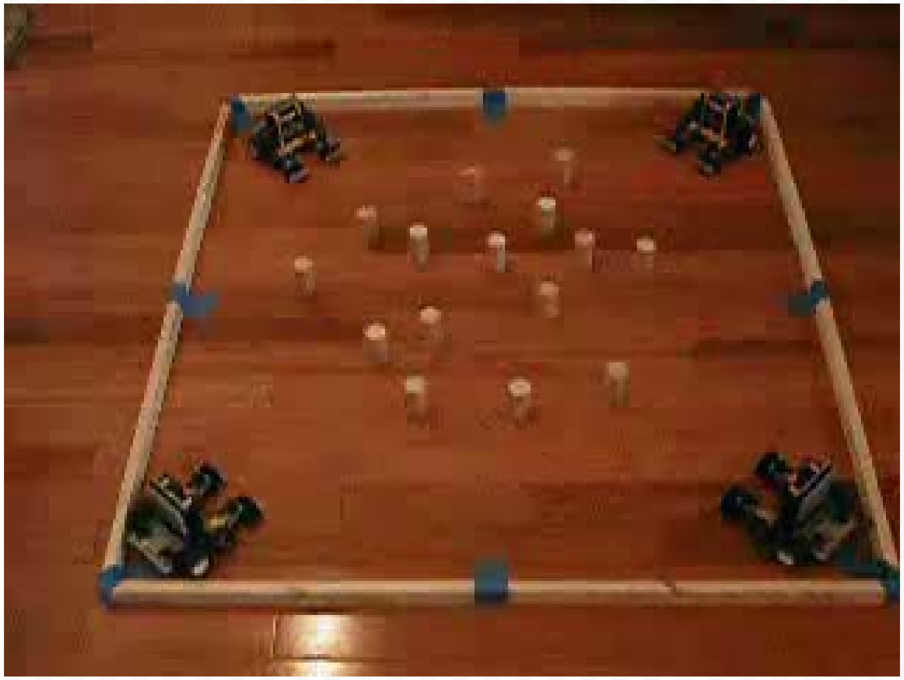
Courses in complexity and systems engineering

Courses in Complexity

Systems of Systems

3 days

- Complexity theory and its application to very complex, networked systems – Solutions in the areas of architecting, integration, collaboration, and T&E



Upcoming Course Schedules

- ***Systems of Systems***
- ***10-12 Sep 07 – Los Angeles, CA***
- 27-29 Nov 07 – Washington, DC
- 26-28 Feb 08 – Orlando, FL

Complexity in Systems of Systems

Thank you for the opportunity to talk about complexity!

Questions?

- ***Systems Engineering***
- ***Training Courses***
- ***Process Improvement***

Scott Workinger, Ph.D.
+1 (650) 363-0979
sworkinger@hcode.com

Courses:
http://www.hcode.com