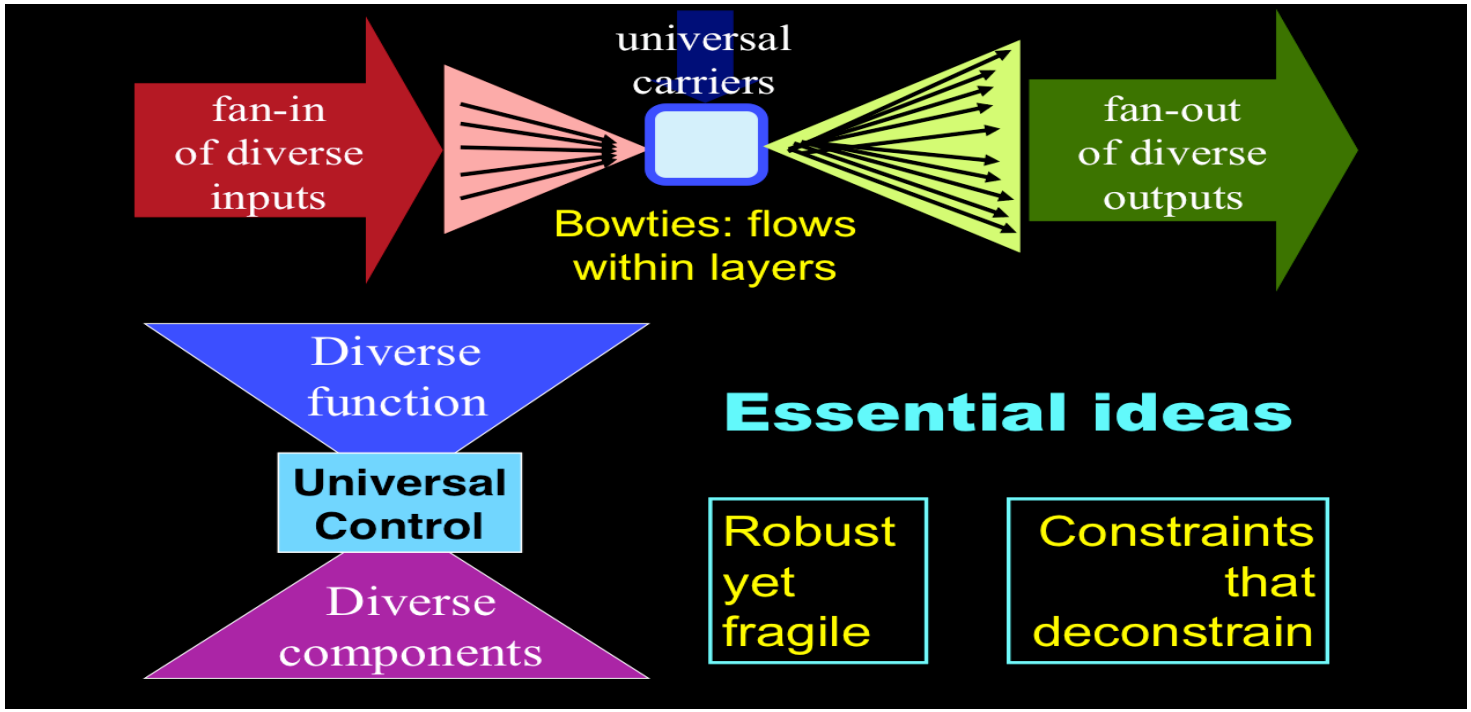
On Understanding Network Complexity (and the direction of the NCRG)



David Meyer

Network Complexity Research Group/IETF 87

dmm@1-4-5.net

http://www.1-4-5.net/~dmm/talks/IETF/87/ncrg/ncrg_direction.pdf

Agenda

- Introduction and Motivation
- Framing the Problem Space
- Who can learn what from who?
- A Bit of Historical Perspective
- NCRG Approach?

Introduction

- This talk is an attempt to frame the NCRG problem space
- My (and others) claim:
  - ***Complex systems such as the Internet can be understood only by identifying their organizing principles, theories, design rules, and protocols***
- The approach advocated here is more akin to Systems Biology in that it seeks to find a *complete* computational system model which embodies such organizing principles, …
- → We need a multidisciplinary approach to describing, understanding, and controlling the properties and dynamics of the whole network
  - Requires the integration of information from many sources

What Have We Learned?

- ***Fortunately***: There are highly organized (and common) universal structures (architectural features) underlying biological and technological networks
  - These structures mediate effective tradeoffs among efficiency, robustness, and evolvability with predicable fragilities

- ***Unfortunately***: The theory for the type of distributed and asynchronous global control used in the Internet (or biology) is relatively new (and its natural language is mathematics)
  - Still possible to find insight into candidate universal structures that can be identified by comparing biological and technological systems

Aside: Universal Architectural Features?

- What we have learned is that there are ***fundamental architectural building blocks*** found in systems that scale and are evolvable. These include
  - RYF complexity/tradeoffs (really a behavior)
  - Bowtie architectures
  - Protocol Based Architectures (PBAs)
  - Massively distributed with *robust* control loops
    - Contrast optimal control loops and hop-by-hop control
    - Hidden robust control loops
    - Cellular growth regulation, packet loss
  - Complexity-robustness spirals
  - Highly layered
    - But with layer violations, e.g., Internet, overlay virtualization

Framing the Problem Space

- I argue that our goal is actually to build a "computationally complete" model
  - Learn from the Systems Biology community
  - Needed for some of our stated goals
    - e.g., is this network more complex than that network, and how/why
- However, viewing the system from the cell (device) level is bewilderingly complex
- Even in engineering large scale ICs are not modeled simultaneously at the whole chip or 'device physics' level
  - Instead modeled with a hierarchy of schemes with various resolutions
- We seem to have neither device level nor systems level models

Who can learn what from who?

- What engineers can learn from biology
  - Biological systems are robust and evolvable in the face of even large changes in environment and system components, yet can be extremely fragile to small perturbations
    - Such universally ***Robust Yet Fragile (RYF)*** complexity is found wherever we look…
    - What architectural features enable this RYF behavior?

- What biologists can learn from engineering
  - Our technology (the Internet) is an obvious example of how a protocol-based architecture facilitates evolution and robustness

- What/where is the commonality?
  - Understanding RYF behavior means understanding the most universal, high-level, persistent elements of organization and protocols.

- **Protocols** → how diverse modules interact
- **Architecture** → how protocols are organized (constraints)

A Bit of Historical Perspective

- Warren Weaver contrasted three classes of problems facing science:
  - Problems of *Simplicity*
  - Problems of *Disorganized Complexity*
  - Problems of *Organized Complexity*

- Let's briefly take a look at each of these

See W. Weaver, "Science and complexity," American Scientist, vol. 36, pp. 536–544, 1948.
See also H. Simon, "The Architecture of Complexity", Proceedings of the American Philosophical Society, Vol. 106, No. 6 (Dec. 12, 1962), pp.467-482.

Problems of Simplicity

- **These are the computationally tractable problems**

- Examples: the pendulum as a simple harmonic oscillator; simple RLC circuits; the interaction of two bodies via gravity; and simple Boolean logic circuits as implemented in much digital hardware

- *Updated definition*: A system is simple if it has ***simple questions*** (i.e., models, theorems, experiments, and computations) to which there are ***robust answers***
  - **Simple questions** are those that can be posed using models that are readily manageable and easy to describe, that theorem statements are short, and that experiments are elegant, are easily described, and require minimal interpretation
  - **Robust answers** are those theorems have simple counterexamples or short proofs, algorithmic scale, and simulations and experiments are reproducible with predictable results

Problems of Disorganized Complexity

- **These problems are not computationally tractable but the variables of interest are randomly distributed and as such can be analyzed with statistical methods**

- Canonical example: In billiards, classical dynamics accurately predicts the trajectories of a small number of balls on a table, and expert players can robustly control these trajectories by keeping them relatively simple.
  - As the number of interacting balls increases, robust predictions become intractable, either computationally or for players
  - However, as the size of the table and the number of balls become very large, specific problems involving ensemble average properties actually become easier and more robust, and statistical methods apply, e.g., statistical mechanics

- In essence, what Weaver called disorganized complexity was ultimately a way to extend the "simple" to large ensembles and, in his thinking, was not really about complexity at all [AldersonDoyle2010]

- Chaos, criticality, scale-free networks all fall into this class

Problems of Organized Complexity

- **These problems are neither computationally tractable nor amenable to statistical analysis due to their non-random structure**
- For example, the statistical methods would not apply if someone were to arrange the balls" and their movements in some highly organized manner.
- Examples include life on earth, the Internet, manufacturing and transportation networks, …
- ***I claim our focus should be on problems of Organized Complexity***

So What is *Complexity*?

"In our view, however, complexity is most succinctly discussed in terms of functionality and its robustness. Specifically, ***we argue that complexity in highly organized systems arises primarily from design strategies intended to create robustness to uncertainty in their environments and component parts.***"

See Alderson, D. and J. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 40, NO. 4, JULY 2010

NCRG Approach?

- RG Drafts
  - draft-retana-network-complexity-framework
  - draft-irtf-ncrg-complexity-framework
  - others

- Missing
  - "device physics"
  - "complete" systems/computational view

- Neither document captures the need for a complete (computational) approach
  - Required to answer one of the questions we've been answer, namely
    - Is this "network" more complex that that "network"

- Questions
  - Need for developing a system level view
  - What is our analogy to Systems Biology?

- Developing a Work Plan
  - And should we be developing deeper interaction with the wider complexity community?

Q&A

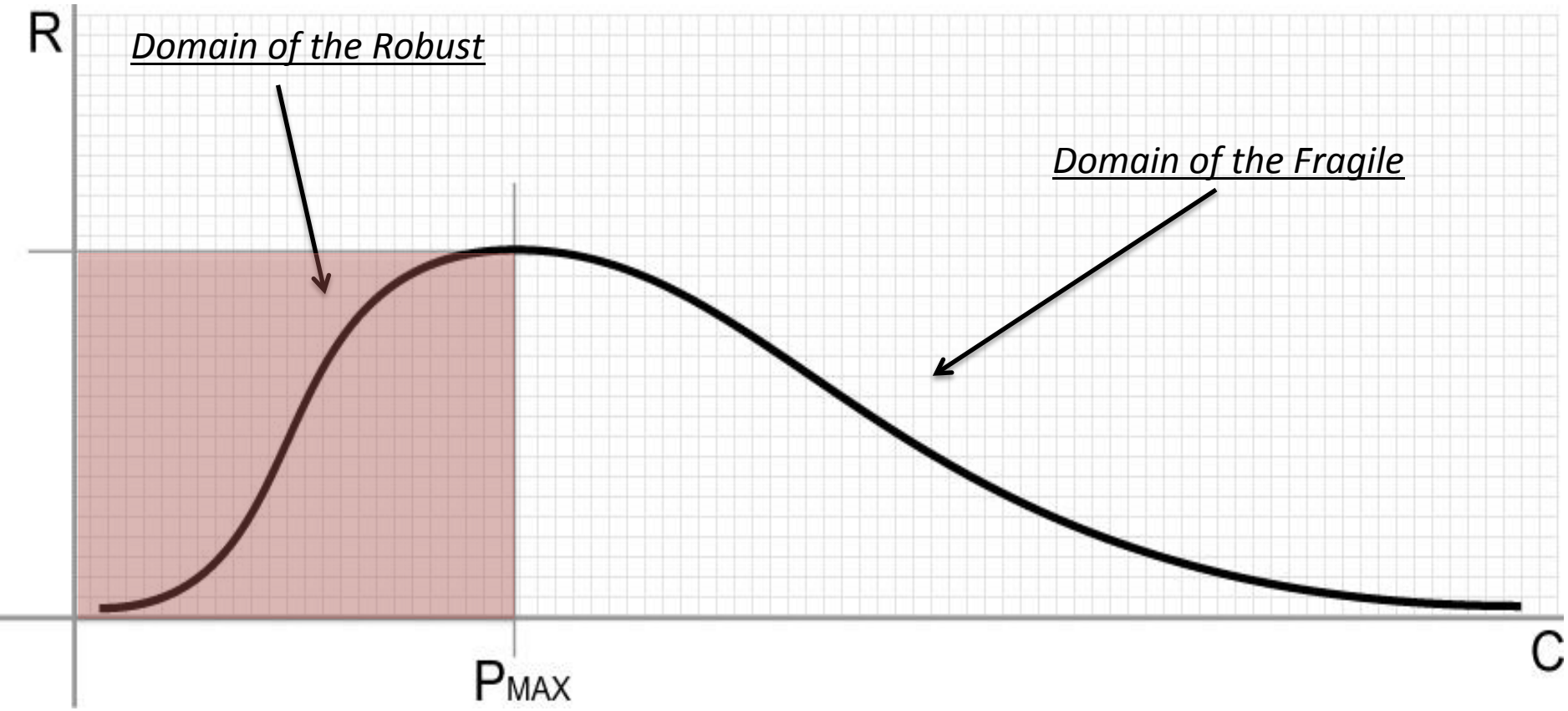
Thanks!

Backup Slides

Universal Architectural Features

- **RYF complexity**
- Bowtie architectures
- Protocol Based Architectures (PBAs)
- Massively distributed with *robust* control loops
  - Contrast optimal control loops and hop-by-hop control
  - Hidden robust control loops
  - Cellular growth regulation, packet loss
- Complexity-robustness spirals
- Highly layered
  - But with layer violations, e.g., Internet, overlay virtualization
- Degeneracy

Robustness vs. Complexity
RYF and Robustness/Complexity Spirals



Increasing number of policies, protocols, configurations and interactions (well, and code)

Can we characterize the Robust and the Fragile?

So what are Robustness and Fragility?

- ***Definition***: A *[property*] of a *[system]* is **robust** if it is *[invariant]* with respect to a *[set of perturbations], up to some limit*

- **Fragility** is the opposite of robustness
  - If you're fragile you depend on 2nd order effects (acceleration) and the curve is concave
  - A little more on this later…

- A system can have a *property* that is *robust* to one set of perturbations and yet *fragile* for a *different property* and/or perturbation → the system is **Robust Yet Fragile (RYF-complex)**
  - Or the system may collapse if it experiences perturbations above a certain threshold (K-fragile)

- Example: A possible ***RYF tradeoff*** is that a system with high efficiency (i.e., using minimal system resources) might be unreliable (i.e., fragile to component failure) or hard to evolve
  - Example: VRRP provides robustness to failure of a router/interface, but introduces fragilities in the protocol/ implementation
  - Complexity/Robustness Spirals

- Conjecture: The *RYF tradeoff* is a hard limit
  - New results show that overall system robustness obeys conservation laws →
  - RYF behavior can be managed but not eliminated

See Alderson, D. and J. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 40, NO. 4, JULY 2010

RYF Examples

Robust	Yet Fragile
☺ Efficient, flexible metabolism	☹ Obesity and diabetes
☺ Complex development	☹ Rich microbe ecosystem
☺ Immune systems	☹ Inflammation, Auto-Im.
☺ Regeneration & renewal	☹ Cancer
Complex societies	Epidemics, war, …
Advanced technologies	Catastrophic failures

- "Evolved" mechanisms for robustness *allow for,* even *facilitate,* novel, severe fragilities elsewhere
- Often involving hijacking/exploiting the same mechanism
  - We've certainly seen this in the Internet space
  - Consider DDOS of various varieties
- There are hard constraints (i.e., theorems with proofs)

Casting System Properties as Robustness

- **Reliability** is robustness to component failures
- **Efficiency** is robustness to resource scarcity
- **Scalability** is robustness to changes to the size and complexity of the system as a whole
- **Modularity** is robustness to structure component rearrangements
- **Evolvability** is robustness of lineages to changes on long time scales

Fragility and Scaling

- A bit of a formal description of fragility
  - Let *z* be some stress level, *p* some property, and
  - Let *H(p,z)* be the (negative valued) harm function
  - Then for the fragile the following must hold
    - ***H(p,nz) < nH(p,z) for 0 < nz < K***

- For example, a coffee cup on a table suffers non-linearly more from large deviations (*H(p, nz)*) than from the cumulative effect of smaller events (*nH(p,z)*)
  - So the cup is damaged far more by *tail events* than those within a few σ of the mean
  - Too theoretical? Perhaps, but consider: ARP storms, micro-loops, congestion collapse, AS 7007, …
  - BTW, nature requires this property
    - Consider: jump off something 1 foot high 30 times v/s jumping off something 30 feet high once

- When we say something scales like $O(n^2)$, what we mean is the damage to the network has constant acceleration (2) for *weird* enough n (e.g., outside say, 10 σ)
  - Again, ARP storms, congestion collapse, AS 7007, DDOS, … → non-linear damage

Quick Aside: What Is Antifragility?

- Antifragility **is not the opposite** of fragility
  - ***Robustness*** is the opposite of fragility
  - Antifragile systems ***improve*** as a result of [perturbation]

- Metaphors
  - **Fragile**: *Sword of Damocles*
    - Upper bound: No damage
    - Lower bound: Completely destroyed
  - **Robust**: *Phoenix*
    - Upper bound == lower bound == no damage
  - **Antifragile**: *Hydra*
    - Lower bound: Robust
    - Upper bound: Becomes better as a result of perturbations (within bounds)

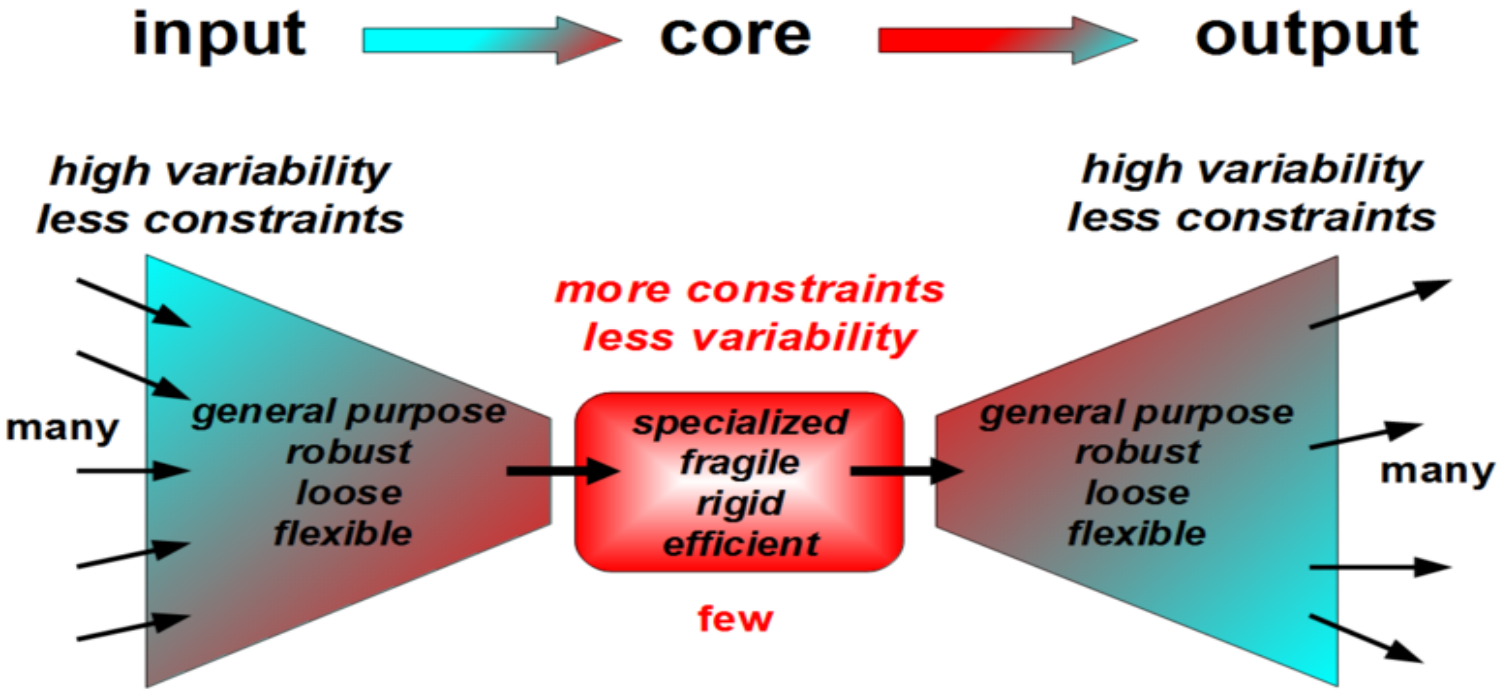
- More detail on this later (if we have time)
  - But see Jim's blog
  - http://www.renesys.com/blog/2013/05/syrian-internet-fragility.shtml

Universal Architectural Features

- RYF complexity
- **Bowtie architectures**
- Protocol Based Architectures (PBAs)
- Massively distributed with *robust* control loops
  - Contrast optimal control loops and hop-by-hop control
  - Hidden robust control loops
  - Cellular growth regulation, packet loss
- Complexity-robustness spirals
- Highly layered
  - But with layer violations, e.g., Internet, overlay virtualization
- Degeneracy

Bowties 101

Constraints that Deconstrain

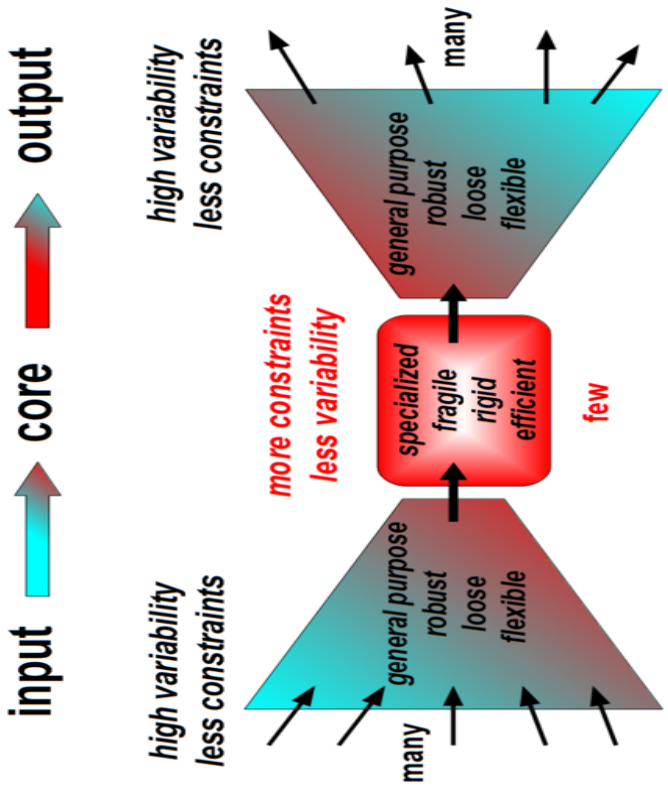


For example, the reactions and metabolites of core metabolism, e.g., *ATP metabolism*, Krebs/Citric Acid cycle signaling networks, …

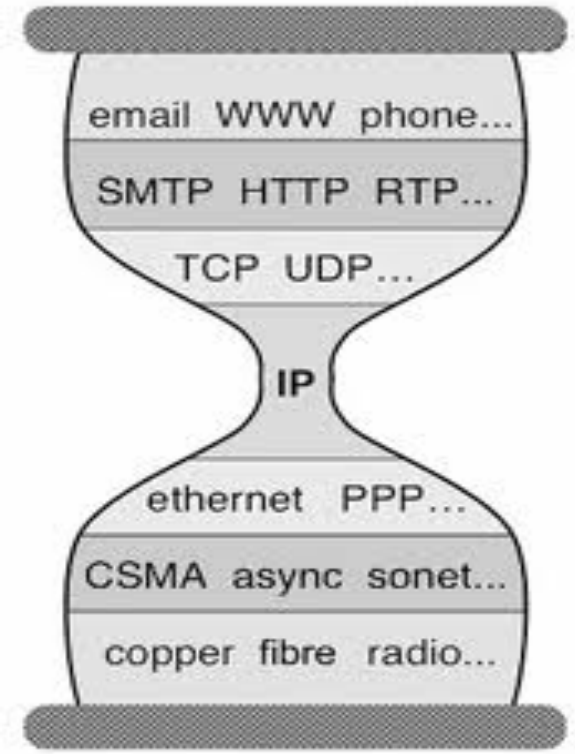
See Kirschner M., and Gerhart J., "Evolvability", Proc Natl Acad Sci USA , 95:8420–8427, 1998.

But Wait a Second

Anything Look Familiar?



Bowtie Architecture

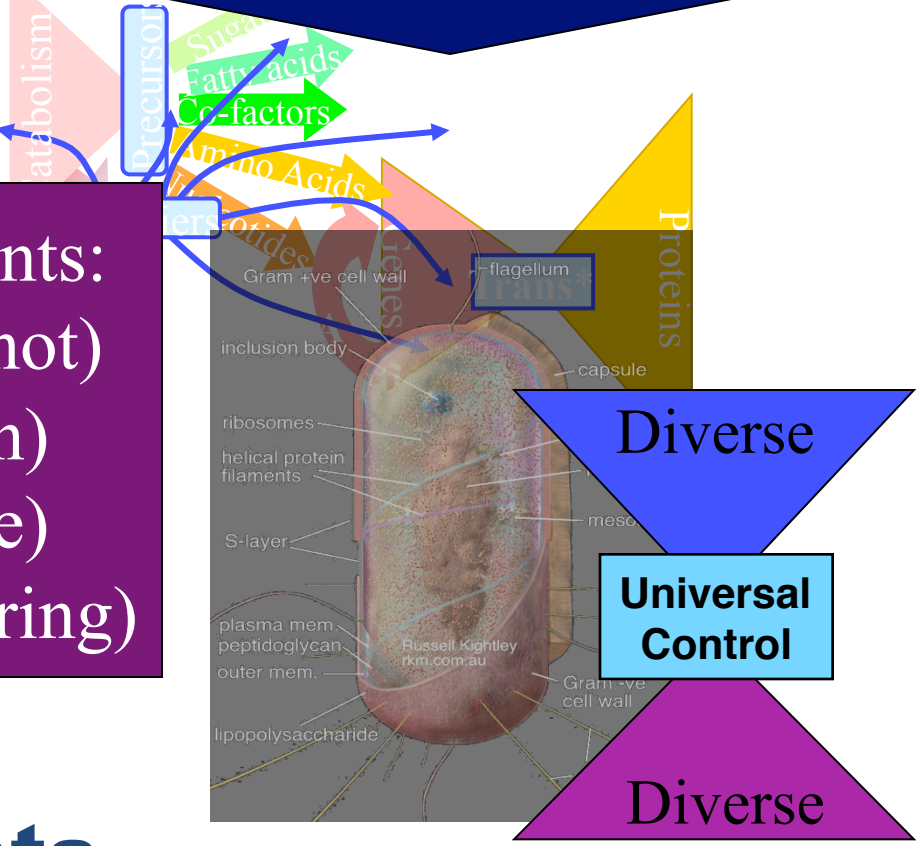


Hourglass Architecture

The *Protocol Hourglass* idea appears to have originated with Steve Deering. See Deering, S., "Watching the Waist of the Protocol Hourglass", IETF 51, 2001, http://www.iab.org/wp-content/IAB-uploads/2011/03/hourglass-london-ietf.pdf. See also Akhshabi, S. and C. Dovrolis, "The Evolution of Layered Protocol Stacks Leads to an Hourglass-Shaped Architecture", http://conferences.sigcomm.org/sigcomm/2011/papers/sigcomm/p206.pdf.

Systems requirements:
functional, efficient,
robust, evolvable

Hard constraints:
Thermo (Carnot)
Info (Shannon)
Control (Bode)
Compute (Turing)

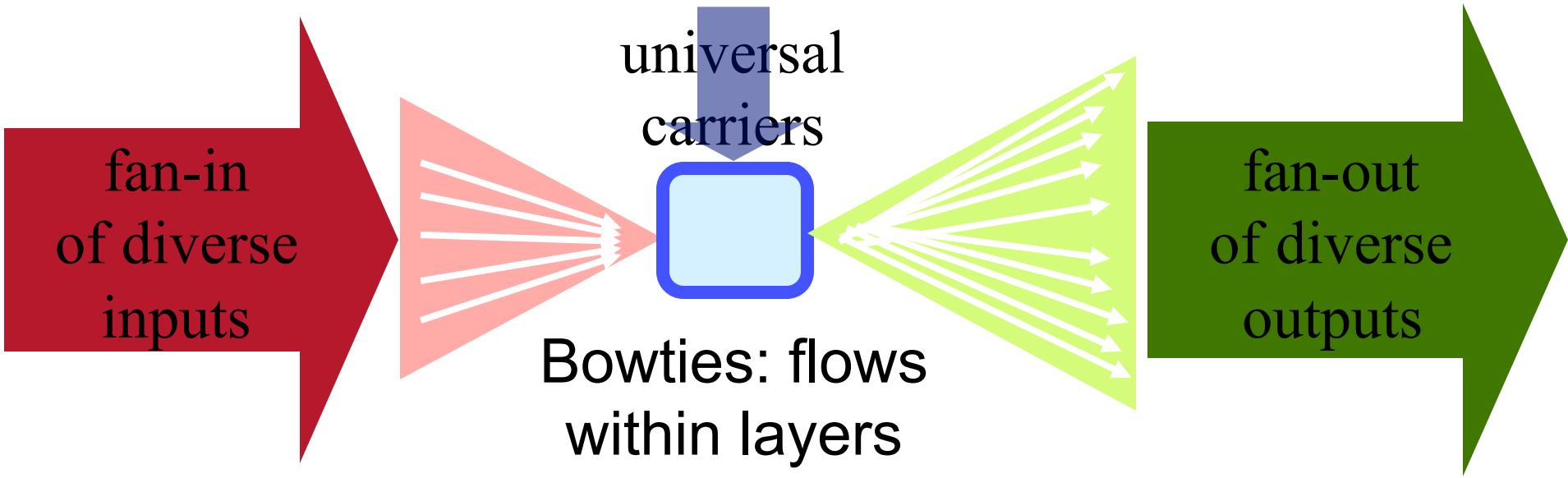


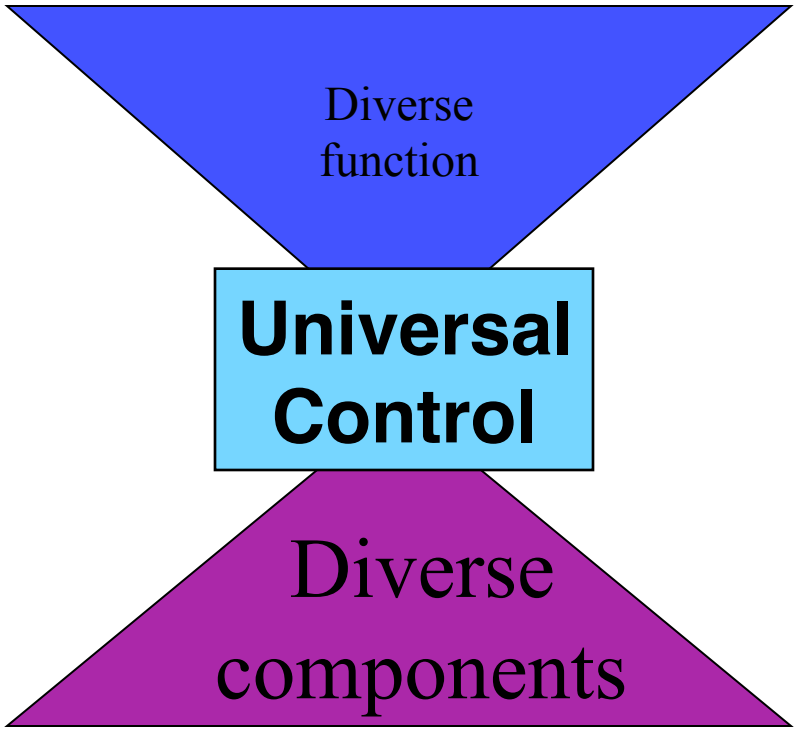
Protocols

Constraints

Components and materials:
Energy, moieties

Slide courtesy John Doyle



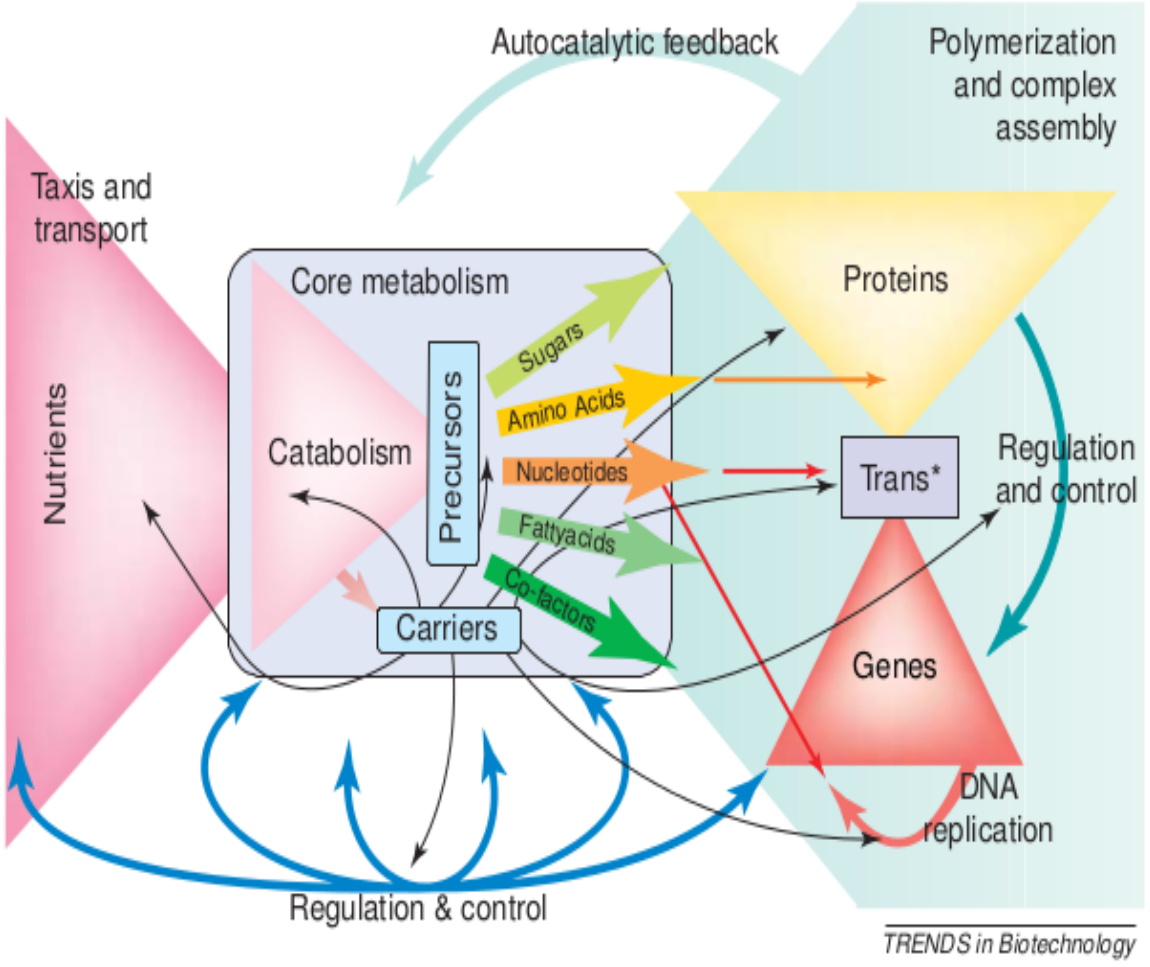


Essential ideas

Robust yet fragile

Constraints that deconstrain

The Nested Bowtie Architecture of Metabolism



See Csete, M. and J. Doyle, "Bow ties, metabolism and disease", TRENDS in Biotechnology, Vol. 22 No. 9, Sept 2004

Biology versus the Internet

Similarities

- Evolvable architecture
- Robust yet fragile
- Layering, modularity
- Hourglass with bowties
- Dynamics
- Feedback
- Distributed/decentralized
- ***Not*** scale-free, edge-of-chaos, self-organized criticality, etc

Differences

- Metabolism
- Materials and energy
- **Autocatalytic feedback**
- Feedback complexity
- Development and regeneration
- >3B years of evolution

What is Complexity, cont…

- The observation is that protocols in such systems organize highly structured and complex modular hierarchies to achieve robustness, but also create fragilities to rare or ignored perturbations
  - Fat tailed distributions: low probability events can have massive consequences
  - → **Protocols are far more important to complexity than are modules**

- The evolution of protocols can lead to a robustness/complexity/fragility spiral where complexity added for robustness also adds new fragilities, which in turn leads to new and thus spiraling complexities

- The most powerful and also dangerous protocols involve feedback control, which also has the most mathematical and thus least widely understood theoretical foundations

- Finally, all of this complexity is largely hidden and deliberately creates the illusion of superficially simple systems, encouraging development of appealing and accessible but completely wrong explanations and theories (bummer)…cf edge of chaos, criticality, scale-free networks

How Complexity Arises

- The specific structure highly organized systems is a consequence of specific ***constraints*** that are placed on their functionality and/or behavior

- Four kinds of constraints
  - component
  - system/environment
  - protocol
  - emergent

- Let's briefly look at each of these

Component-Level Constraints

- The components that comprise any system are typically constrained in terms of what they can do
  - even separately
  - ultrareliability vs. uncertain reliability

- For example, much of mechanics, electrical circuits, chemical processes, etc., can be described in terms of relationships such as F = MA and V = IR
  - Of course, these constraints are often expressed as differential or algebraic equations
  - But also much more general

- The **uncertainty** of components often imposes constraints on a complex system that are as important as the nominal idealized component behavior

System-Level Constraints

- What distinguishes biology and technology from other types of systems is that there are complex constraints on the system as a whole that **are not consequences of those on the components**, including functional requirements,

- These include
  - What the system needs to do
  - Environmental and operating requirements required to achieve robustness
  - Robustness to uncertainty and perturbations from the environment

Protocols

- Protocols allow communication with the external environment
- Protocols typically take the form of rules for the configuration and/or interaction of system components, may impose additional constraints on the overall system
- Although these additional constraints may reduce the number of possible system solutions, a “good” set of protocols minimally constrains these solutions so as to facilitate a focus on the feasible and robust solutions

Emergent Constraints

- The interaction of the previously mentioned constraints can imply an additional set →
- "Emergent" constraints that are **nontrivial consequences** of the interaction between the system and component-level constraints, and possibly protocols
- Perhaps the most important emergent property of any set of constraints **is whether their intersection is (non)empty**, so theory and methods to determine this are central to engineering specification and design
- Emergence is also associated with ***unintended consequences*** for either good (an emergent benefit) or bad (an emergent fragility).

So What is *Architecture*?

- Architectures are fundamentally about handling or creating the various constraints described previously to facilitate "good" solutions among competing tradeoffs
  - Can be designed, evolved, or both
  - draft-retana-network-complexity-framework starts to talk about tradeoffs in an informal way

- No coherent theory of architecture (yet)
  - However, "architecture is about how modules are connected via protocols"
  - "Constraints"

Protocol Based Architectures (PBAs)

- Protocols are a basic part of organized complexity
- Gerhart and Kirchner [1] nicely capture the role of protocols in the biological domain with the phrases "constraints that deconstrain" and "facilitated variation."
  - They describe how constraints in the **form of universal shared protocols** provide a platform for diverse functionality and robustness by *facilitating* large but functional *variation* on which selection can act
- PBAs allow typical network behavior to be fine-tuned through elaborate but hidden control systems and thus appear boringly robust despite large internal and external perturbations
- For *PBAs,* the *protocols* (rules of interaction that persist) are more fundamental than the *modules* (which obey protocols and can change and diversify)

[1] Gerhart, J., and M. Kirschner, "The theory of facilitated variation", http://www.pnas.org/content/104/suppl.1/8582.full.pdf

RYF in PBAs

- Bacterial and Internet PBAs underlie both their robustness and fragility
  - PBAs allow typical network behavior to be fine-tuned **through elaborate but hidden control systems** and as such appear boringly robust despite large internal and external perturbations
- PBAs also facilitate evolution
  - from microbes to humans
  - from an academic research network to a global information infrastructure
- An important result of layering and control is that complexity and fragility remain ***largely hidden***, often revealed only by catastrophic failures and often only after the system has absorbed multiple insults without apparent consequences
  - Remember this for the discussion on non-linearity
  - Is the cumulative effect of the "multiple insults" related to RYF behavior?
- Basically, large structured rearrangements are tolerated by control systems that reallocate network resources, easily conferring robustness to outright failures of components and brute force attacks,
  - However, cost of this robustness is disastrous fragilities to small random or targeted changes that subtly violate core protocols
- The greatest fragility in PBAs is from parasites and predators, who hijack and consume universal and standardized interfaces and building blocks