ESD.84 Doctoral Seminar – Session 5 Notes
Guests Presenting: John Doyle and Dan Hastings

Initial Session Design:

- Welcome and Overview and Introductions (5-10 min.)
- Generating Key Questions from Readings (10-15 min.)
- Perspectives on Complex Systems – John Doyle (30-45 min.)
- Principles of System Architecting-an Overview – Ben Koo (15-20 min.)
- Break (15 min.)
- Technology Policy and Implementation – Dan Hastings (15-20 min.)
- Book Reviews (5-10 min. each):
  - William Baumol, *The Free-Market Innovation Machine.* Princeton: Princeton University Press (2002) -- Marcus Sarofirm
  - Ulrich & Eppinger, *Product Design & Development*, selections – Eun Suk Suh
  - Christensen, Clayton, *Innovator's Dilemma* – Chris Musso
- Integrative Discussion (20-30 min.)
- Next Steps (10-15 min.)

Initial Brainstorming of Questions:

- How to build systems that are more robust to link damage – where the systems collapse at these critical links?
- How can we learn more about what comes before the dynamics of firms competing in industry – there is less on the emergence of structures in the first place?
- With system increasing system complexity over time is there also increasing system fragility?
- Boeing 777 as a living example in relation to living systems – how far can or should we go with the analogy between mechanical and biological systems?
- How to think about fragility and uncertainly with additional forms of uncertainly beyond known unknowns? There are specific fragilities that are known when you design a system – but how to think systematically about things you don't know?
- How to extend the implications from biological metaphors to social systems – to organizational issues?

Discussion:

It is much harder to synthesize than it is to analyze

- Does there exist a design that is “okay” for all uncertainty?
- Key break through in verification
- How to systematically take big systems and understand their fragilities?
- It requires a model of what can go wrong?
- Major break through in how to search large space of potential things that can go wrong – real semi-algebraic geometry and non-self adjoint operator theory
- Even if you have a good model of a system, there is a combinatorial problem still – but there is the larger issue of the limits of the model and its assumptions
- The difference between rigid and fragile is not just a single parameter – these are two different assumptions that can play out in different ways depending on how sensitive the system is to which is the characteristic
- The mindset is that most parameters have the property of “I don’t know” and there is an iterative modeling and development process as you do different experiments to address the “I don’t knows”
- There are some dimensions that are robust and can be modeled in a sloppy way and it is okay – while some require extremely sensitive models – it is an iterative process of determining which is which
- Searching combinatorially large spaces (global optimization of non-linear convex functions) in polynomial time -- has been a key breakthrough – new mathematics is needed here and there have been important breakthroughs
- Non parametric uncertainty in models is also a key area of breakthrough
- The contrast between pre-calculus physics and physics with calculus – that is where we are with complex systems
- Constraints and optimization versus provability and reachability – the complexity of the computation scales up but the questions you ask remain simple (“nothing bad happens” as a simple specification with complex implications)

The scope of biological metaphors

- Darwinian evolution is the only mechanism
- It is a myth that most of biological evolution is trial and error – there are highly non-random response to environmental changes – this is a much more “engineered” process than we anticipate – the key is the power of feedback
- It is also a myth that engineering is not Darwinian – we evolve and mutate on a given problem
- The two have very different substrates
- There are only certain architectures that produce robust systems
- Evolving complexity – size is a surrogate for complexity in the fossil record and history full of shifts from very large to very small to very large iterations – exponential growth, catastrophic failure, exponential growth, etc.
- We are large – in that our buildings and technology looms large
- Asteroids are rare events that are only catastrophic if you are fragile to it
- It is necessarily a monotonic relationship between complexity and fragility – fragility is a conserved quantity – a theorem – though the math is still in its infancy
- There are background catastrophic events that you can do nothing about – such as super novas – so you should reduce the probabilities of the fragilities to below those of the catastrophic events
- Putting the most critical technology in a fighter plane under the pilot’s seat since losing the pilot loses everything anyway
- Robustness and fragility are the two primary drivers of complexity – the first order design criteria (robustness is what drives increased complexity which then creates new fragility which then drives more robustness which then drives more complexity – a spiral relationship that requires active management

Conservation of fragility

- Consider the room – it has walls and it also has a thermostat – much more complex and better when it works – but the system is more fragile if the thermostat goes wrong
- Move fragilities into the cheap small high performance bits that are controllable rather than in the high energy large aspects of systems
- Relationship between flexibility and robustness – flexibility is a mechanism to adapt to different disturbances, but the sensors, actuators and software that give you flexibility also add fragility
- Evolvability as a word rather than flexibility – this is just robustness on a different time scale

The price of metaphor is vigilance

- Do the math first and then do the metaphor
- Too many of the metaphors are wrong – complexity is not at the edge of chaos, etc.
- The math that we have developed to deal with biology and networks does help with many key physics problems
- Still we should see parallel patterns in the robust properties of very different complex systems – same trade-offs between robustness and complexity
- Autocatalytic properties of social systems are much more difficult than regulatory social system
- Fundamental question in the value of exponentially growing aspects of our economy – introduces new fragilities
- Robustness and fragility are inverses of each other – builds on linear control systems
- Focusing fragility to the areas where you can respond – a key guiding principles in designing robust systems (in addition to "when in doubt, make it stout") – better to build in the fragilities and know where they are – if you don't you are in denial (a fuse in an electrical system – though there can then be cascading power events initiated by multiple local failures)
- This is what civil engineers do – not just extra concrete, but also indicators
- Key challenge of the unknowability of the interconnections
- Lesson is to concentrate on making yourself robust to your assumptions – which is back to the iterative learning process (look at dual variables as sensitivities to your constraints)

Presentation by Ben Koo

- Organization of energy, matter and information as the key work a system architect
- Bridging from semantics to syntax to pragmatics
- design parameters → functional requirements → customer needs → process variables
- HTML as a fundamental protocol break-through in software architecture
- Changing a protocol is like running a program
- Architecting protocols – power of knowing which protocol to change – a unique perception on the part of architects – how to see the emergent properties before you have created the system
- Principle 1: Architecture intent should be encoded in statements of needs and requirements
    - A subjective matter initially – selection of language and protocols is central
- Principle 2: Requirements statements are iteratively modified based on system feedback
    - Changes in architectural requirements redefines architectural space
- Principle 3: The dynamic evolution of architectures follows the least resistance path
- Architecture is an instrument for intent preservation and expression

Discussion:

- The history of computer science did not focus on research about protocols and then they became almost laws of nature such as windows – so no research because it is so obscure or so dominant
- The choice of going to the moon actually triggered a debate on the merits of getting to the moon directly or first to a space station around the earth – the latter had more certainty but a larger upside potential

Presentation by Dan Hastings:

- Contrasting time scales in government and industry on science and technology investment
- Range of investment in R&D is between 1% and 16% of sales across different industries – 16% is pharmaceuticals driven by patent expiration while the concrete industry is around 1%
- Air Force concluded the range is between 2% and 2.4% of total obligation authority
- Focus on the technology choices around night vision, GPS and F-22 – key lessons on the inability to see the full value of GPS in advance

Discussion:

- The increasing capability of smart bombs now reveals new fragility around the requirements for more precise information on appropriate targets
- The mix of top-down guidance and bottom-up input
- What about incentives – free market mechanisms or others?
- Shifting nature of the organizational structure to be more responsive to the warfighter – an important shift but there is still the issue that this would still not give you GPS
- There are systematic architectural features of the success stories – especially looking for systems that empower people who are different from the dominant decision makers
    - Sensors always end up driving actuators that you didn't anticipate
    - Build good sensor systems and the actuators will come
    - The key is to get the protocol in the waist of the hour glass right and the upper and lower parts will vary
- Waists of hour glasses are the new commons – but we are allowing them to be copyrighted and limiting use
- Innovation is not the widgets – it is the protocols

Book review of Steve Eppinger's book on Product Design and Development by Eun Suk Suh

- Dividing product development into six different stages
    I. Planning
    II. Concept Development
    III. System Level Design
    IV. Detail Design
    V. Testing and Refinement
    VI. Production Ramp-Up
    VII. Project Management
- A useful overview of the entire product development process – able to see the forest, not just the trees
- Issue of iterations with a linear process – what happens when you get to the end and don't like what you see?
- Missing element – verification and validation (real feedback on what will really work) – it is more than robust engineering