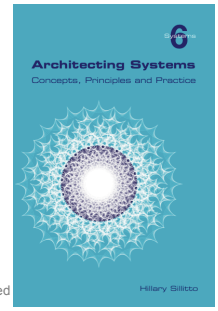
Challenges of the Third Age of Systems Engineering - "open systems on a closed planet"

Hillary Sillitto, Ceng, FInstP
ESEP, INCOSE Fellow
Visiting Professor, University of Bristol Systems Centre
Former Thales UK Director of Systems Engineering
Past President of UK Chapter of INCOSE (2004-6)
Author of *Architecting Systems: Concepts, Principles and Practice*



Scheme of talk

- "Ages" of SE
- Everything is interconnected
- Increase in complexity and scale
- Architecting Systems
- Systems issues facing the systems communities and society
- INCOSE 2025 Systems Engineering Vision
- What next?

Prehistory: from the Pyramids to ca 1950
↓
Closed systems in an infinite environment
↓
Open systems in an infinite environment
↓
Open systems on a closed planet

Principal sources:

- My own work
  - *Architecting Systems: Concepts, Principles and Practice*
  - College Publications, 2014
- Transformational Systems Engineering Caucus
  - (acknowledged as Scott Workinger)
- INCOSE SE 2025 vision
  - (acknowledged as Copyright © 2014 by INCOSE)

The Ages of Systems Engineering

- Prehistoric: from the stone age to ~1950
  - Pyramids – 4 Millennia ago!
  - Warships, from before Napoleonic era
  - WW2 networked Command, Control & Intelligence systems
    - Air Defence, Enigma, ULTRA….

The First Age of Systems Engineering

- SE Recognised as a distinct activity in 1950s
- Hugely ambitious, risky, unprecedented projects
  - ICBMs, Nuclear submarines, Apollo programme
  - Clear boundary, limited interactions with rest of world

Closed systems in an infinite environment

Subsystems
Subsystems
Subsystems

Product systems

The Second Age of Systems Engineering - from ca 1990

- Systems were interconnected, interacting, interdependent
- The internet, ad-hoc coalition military operations…

(And by the way, please deliver to time and budget…..)

Closed systems in an infinite environment

Open systems in an infinite environment

(Derek Hitchins' 5-level classification)

Subsystems & technical artefacts | *Project systems* | *Business systems* | *Industry systems* | *Societal systems*

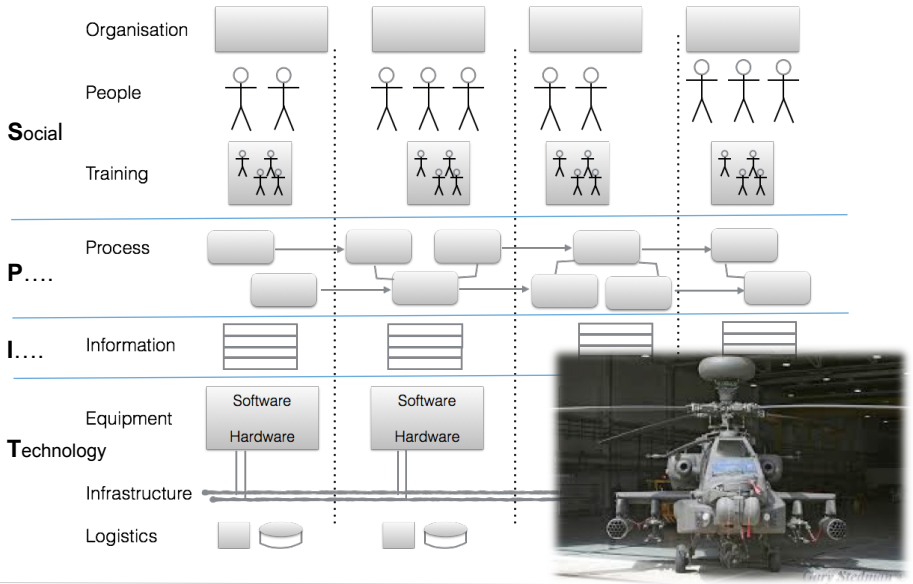
Subsystems
Subsystems
Subsystems

Product systems → Capability → Service → Enterprise

€£$ value

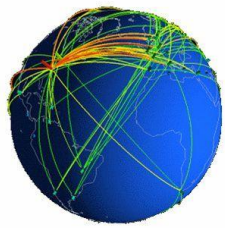
Societal value

"Capability Engineering" – orchestrating all the "components of capability" needed to make effective use of a product system

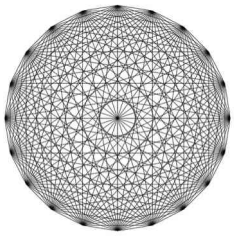


The Paradigm Shift in Systems

- Networked Computation affects:
- The systems we develop, test, evaluate & manage:
  - Larger
  - More complex
- The systems we use to test & support them:
  - Often Network Based
  - Test Environments are often Systems of Systems
- Integrated legacy systems
  - "Green field" designs → less frequent
  - Capability engineering → becoming common
- Organizational relationships are changing
  - Top-down policy directives → reaching limits of effectiveness.
  - Agile Engineering & Spiral development → often preferred
  - T&E → extending through lifecycle
  - Stakeholders →large, diverse groups
- Timing
  - Working on "Internet time" → development & testing schedules compressed
  - Capability development extends through lifecycle



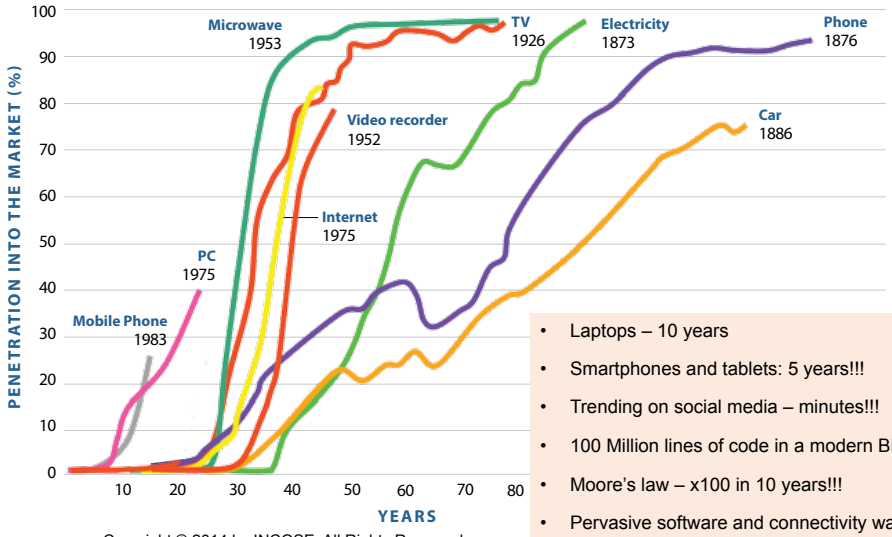
Network Effect



Metcalfe's Law
$V = k * N^2$

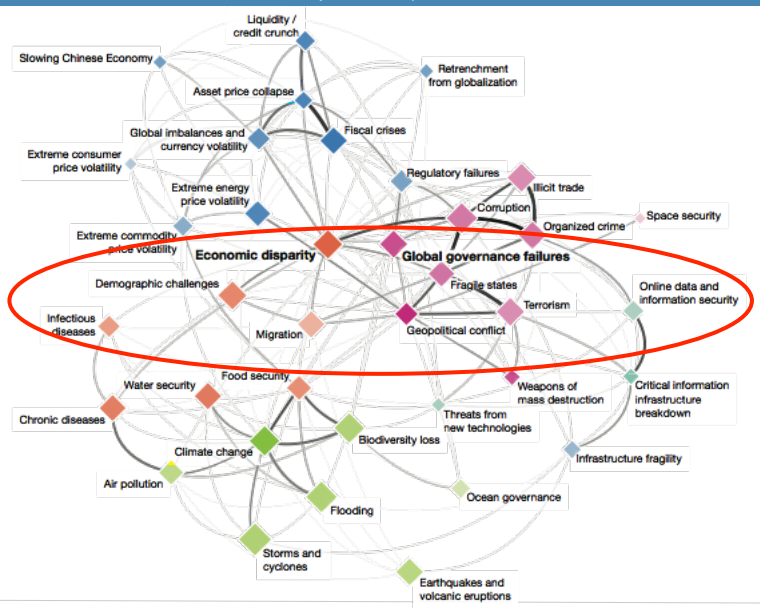
ScottWorkinger@gmail.com

Unprecedented rates of change: time to market saturation



- Laptops – 10 years
- Smartphones and tablets: 5 years!!!
- Trending on social media – minutes!!!
- 100 Million lines of code in a modern BMW
- Moore's law – x100 in 10 years!!!
- Pervasive software and connectivity was THE BIG CHANGE in the last 30 years
- Environmental sustainability and resilience is THE BIG CHALLENGE for the next 30

World-scale systemic interactions
(from World Economic Forum Risk report 2011)



The Third Age of Systems Engineering is upon us!

- Man-made systems span the planet
- Planetary scale interactions – planetary scale emergence – planetary scale risks
- Neither the planet, nor our brains, have got any bigger in 4 millennia –
  - and not wired to handle complexity and delayed effects –
  - intuition doesn't work, need conscious thought

Closed systems in an infinite environment

↓

Open systems in an infinite environment

↓

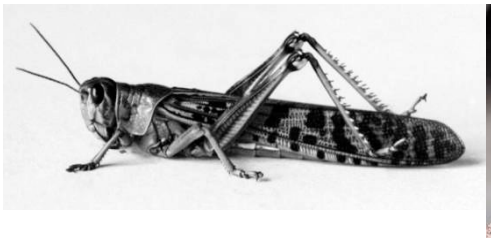
Open systems on a closed planet

Unintended consequences if you choose the wrong boundary

- Mao's Sparrows

In my hutong (neighbourhood) in Beijing, by contrast, the mornings were strangely silent.
In 1958 Mao Zedong had declared war on songbirds, sparrows in particular: he claimed they consumed scarce grain.
For three days and nights my neighbourhood, gripped like much of northern China by hysteria, had beaten pots and pans to keep birds on the move until they collapsed with exhaustion on the roofs and pavements of the courtyard houses.
The consequence was a plague of locusts the next year that helped bring on a famine.
"Suan le", Mao had said when told that the anti-sparrow campaign was not working. "Forget it then".

- The Economist, December 20th 2008, "The loneliness of the Chinese birdwatcher"
- Photos from Wikipedia



Coping with complexity

FROM	TO
Today, stakeholders are demanding increasingly complex systems yet complexity-related system misunderstanding is at the root of significant cost overruns and system failures. There is broad recognition that there is no end in sight to the system complexity curve.	In 2025 and beyond, complexity will be an objective system metric, and methods for minimizing unnecessary complexity as well as understanding and measuring complex systems will be commonplace

Needs:

Both formal and semi-formal methods for identifying emergent and unanticipated behaviors.

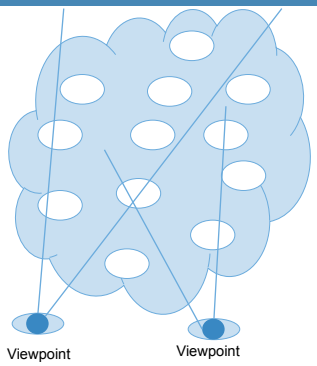
Analytical techniques to explore huge system state spaces to identify and eliminate undesirable system states.

Techniques to correlate a diverse range of system parameters as indicators of system health.

Capitalizing on this understanding to develop systems that are more fault tolerant, secure, robust, resilient, and adaptable will be a fundamental part of systems engineering practices.

Generic model of complex situation

Complex situations are often
a) misunderstood
b) under-conceptualised
c) poorly managed



1. The view from a given viewpoint is restricted.
"What can be seen"

2. Observation by particular observer is selective and prone to error.
"What is seen"

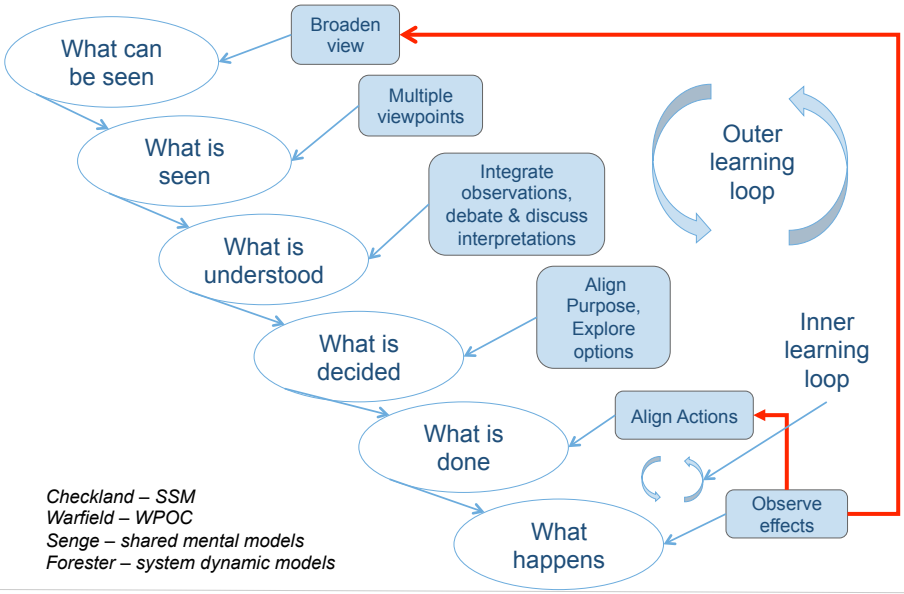
3. Interpretation of the observation is conditioned by the observer's mental model.
"What is understood"

4. Decision depends on what is understood, what is important, perceived affordances, anticipated effects and side-effects.
"What is decided"

5. Action is the actual intervention (as opposed to the intended one) and how it interacts with the real (as opposed to perceived) system.
"What is done"

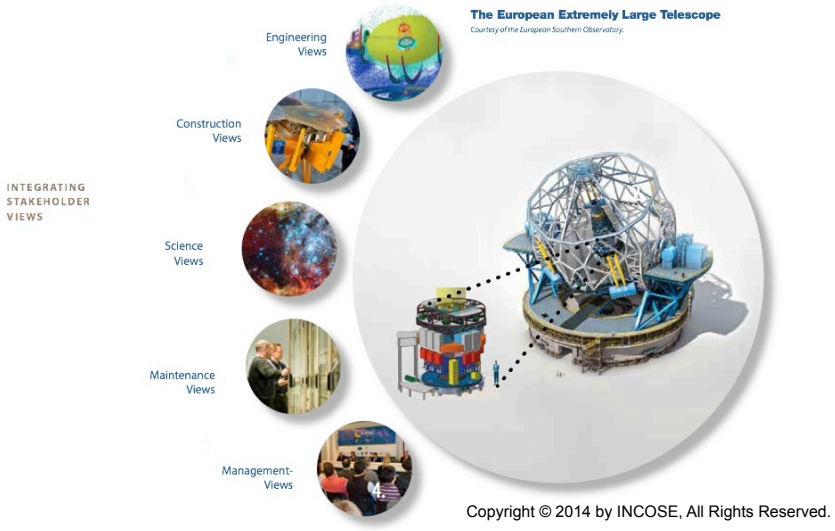
6. Effects of action – short and long term, actual and perceived
"What happened"

Double Loop Learning improves our ability to deal with subjective complexity



Checkland – SSM
Warfield – WPOC
Senge – shared mental models
Forester – system dynamic models

Architecture – multiple stakeholder viewpoints



Goal of Architecting[1]

The purpose of "systems architecture" is to ensure that the various parts of our systems, when connected to each other and placed in their operating environment:

- fit together
- work together
- achieve the required effect
- do not produce unacceptable side-effects

and can be

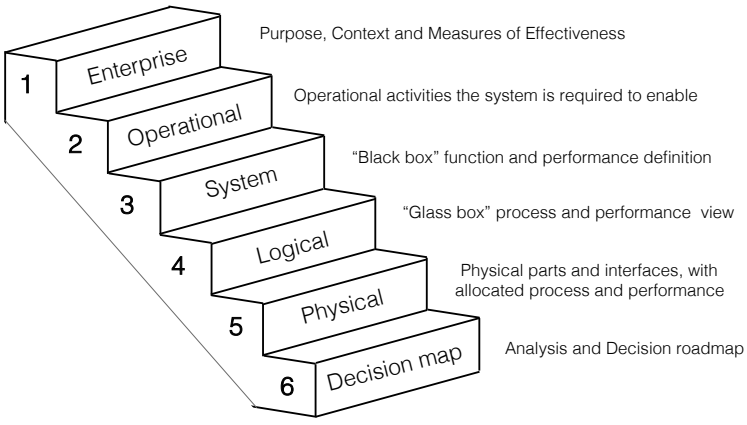
- kept operational over time
- reconfigured to meet "reasonable unforeseen" circumstances.

[1] Sillitto, H G, "Composable Capability – Principles, strategies and methods for capability systems engineering" – INCOSE International Symposium, Philadelphia, June 2013

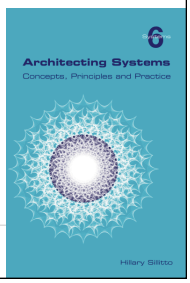
What is "Architecting"?

- "Structuring for success" – (in product world, equates to initial or high level design)
- Creative not mechanistic
- Art as well as science – blend intuition and analysis
- Look for short cuts through immense option space based on
  - Patterns
  - Heuristics
  - "what worked before" - but what about unprecedented systems? – (look for analogies, general system patterns & principles)
- Good architects do it naturally – learn from own and others' mistakes
  - Methods, mindset and approach can be learnt
  - Need to create environment where mistakes are affordable and feedback is fast!
- Key skills
  - Abstraction
  - Granularity
  - Boundary
- Good architecture
  - "Obvious" - right level of abstraction
  - "Simple" – right level of granularity
  - "Shows how it all fits together (and works together)" – good choice of boundaries, clean allocation of functions to physical elements
  - Coherence and consistency across the solution - gives integrity, resilience, adaptability, scalability….

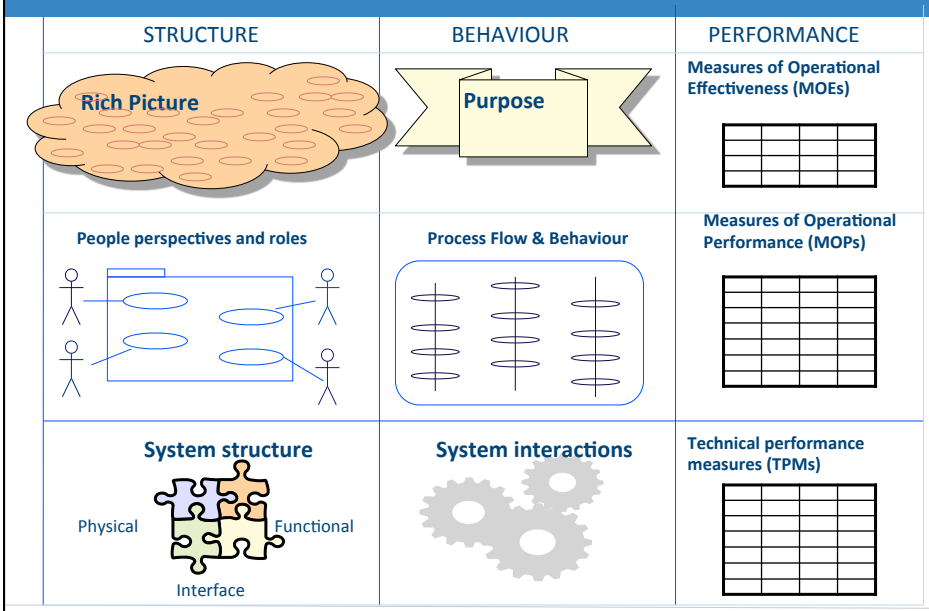
How to architect?
6 Key architecting perspectives – develop in 6 step architecting process[1]



[1] Sillitto, H G, *Architecting Systems: Concepts, Principles and Practice* – College Publications, 2014



Managing the information: a simple architecture framework



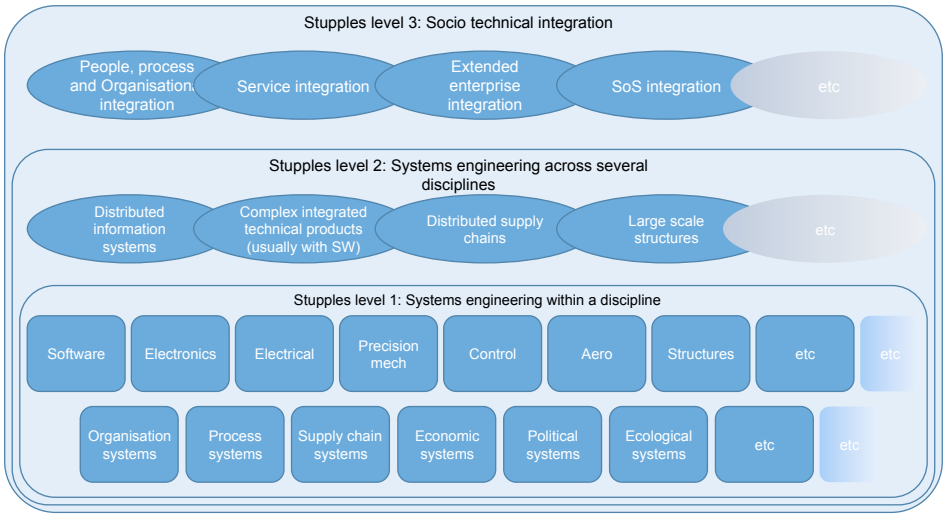
Barriers to getting best value from SE effort

1. Mission complexity is growing faster than our ability to manage it . . . increasing mission risk from inadequate specifications and incomplete verification.
2. System design emerges from pieces, rather than from architecture . . . resulting in systems that are brittle, difficult to test, and complex and expensive to operate.
3. Knowledge and investment are lost at project life cycle phase boundaries . . . increasing development cost and risk of late discovery of design problems
4. Knowledge and investment are lost between projects . . . increasing cost and risk: dampening the potential for true product lines.
5. Technical and programmatic sides of projects are poorly coupled . . . hampering effective project decision making; increasing development risk.

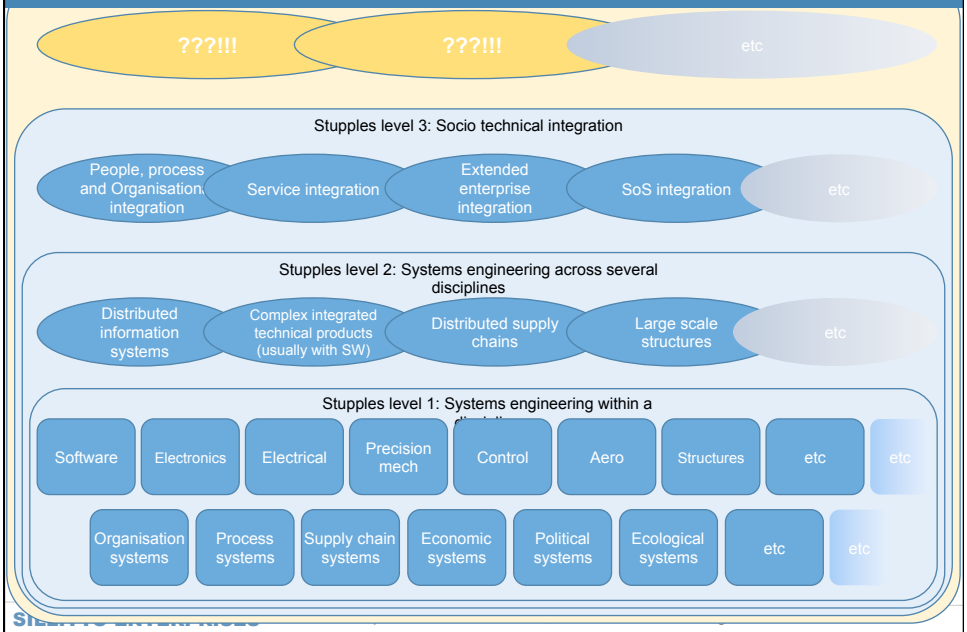
6: Abstract Information Structures and the methods needed to develop them don't fit most practitioners' and managers' mental models

The breadth of systems engineering competencies

Systems Integration – within a discipline – across disciplines – sociotechnical (Stupples, adapted by Elliott et al, graphic from Sillitto 2011)



What about Environmental Systems Integration?

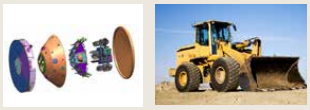


Vision for future, and a possible approach

ADAPTABLE AND SCALABLE METHODS

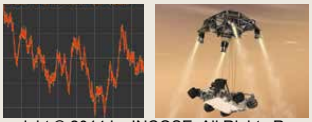
Systems engineering methods will be scalable to system and organizational complexity and size, and tailored to the application domain. Method selection and adaptation will be value driven to optimize project schedule, cost, and technical risk. Methods and tools will scale from small and medium sized enterprises to multi-billion dollar projects.

TAILORED TO THE DOMAIN



SCALED TO PROJECT SIZE

SCALED TO SYSTEM COMPLEXITY



"TSE" (Transformational Systems Engineering) Caucus – an initiative by INCOSE Silicon Valley Chapter

- Examining new practices and their inherent opportunities:
  - Agile
  - Complex Systems
  - HSI
  - MBSE
  - SoSE
  - Design Thinking
  - Systems Science
  - …

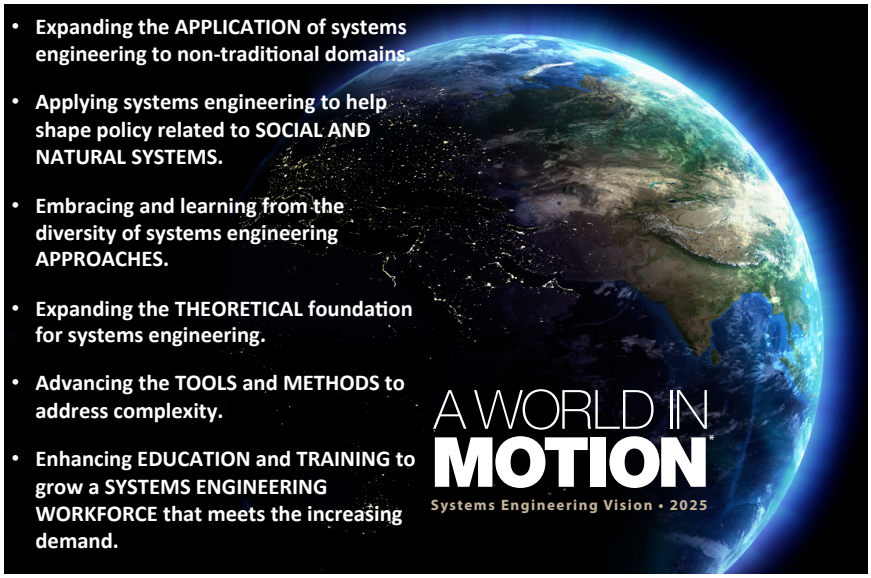
- Designing for Emergence in Systems Engineering, itself

"*… it is in many ways an even bigger effort, and one that may never be actually 'done'. Perhaps what we are doing is setting a multi decade effort into motion to create a systems engineering practice that will evolve with the environment.*" - *Lee Amon*

ScottWorkinger@gmail.com

INCOSE SE Vision 2025: 6 imperatives

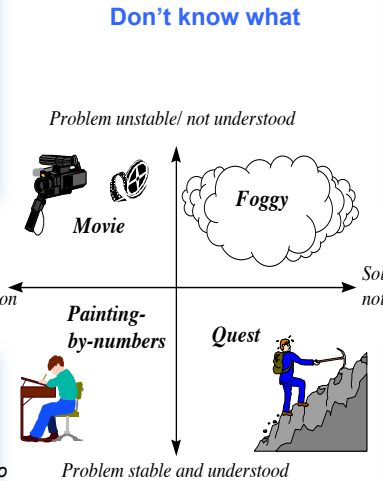
- Expanding the APPLICATION of systems engineering to non-traditional domains.
- Applying systems engineering to help shape policy related to SOCIAL AND NATURAL SYSTEMS.
- Embracing and learning from the diversity of systems engineering APPROACHES.
- Expanding the THEORETICAL foundation for systems engineering.
- Advancing the TOOLS and METHODS to address complexity.
- Enhancing EDUCATION and TRAINING to grow a SYSTEMS ENGINEERING WORKFORCE that meets the increasing demand.



Different strategies for different situations

Movie:
Know how but not what.
Explore problem space using standard process.
Offer multiple solutions:
"I'll know it when I see it"

Foggy:
Don't know what or how.
Combines "Movie" and "Quest".
Usual for breakthrough products: - market don't know they want it, developer not sure what's possible

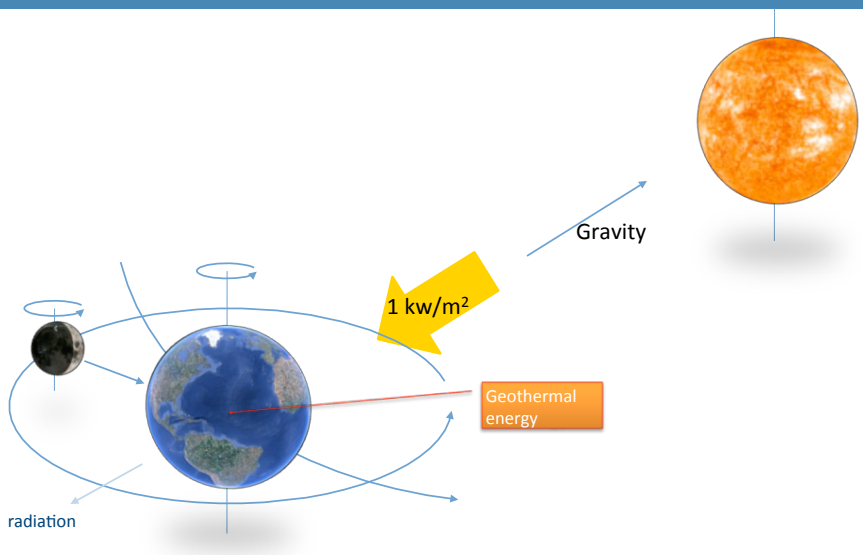


Painting by Numbers:
Know what and how.
Define – build – sell.
This is where you need to get to for series manufacture.

Quest:
Know what but not how. Usual for technology innovation.
E.g. "get a man to the moon and back this decade".

Reported at INCOSE 1st European Systems Engineering Conference, based on Obeng E, *All Change! Project Leader's Secret Handbook* (Financial Times/Prentice Hall, 1995)

Thank you for your attention Questions?

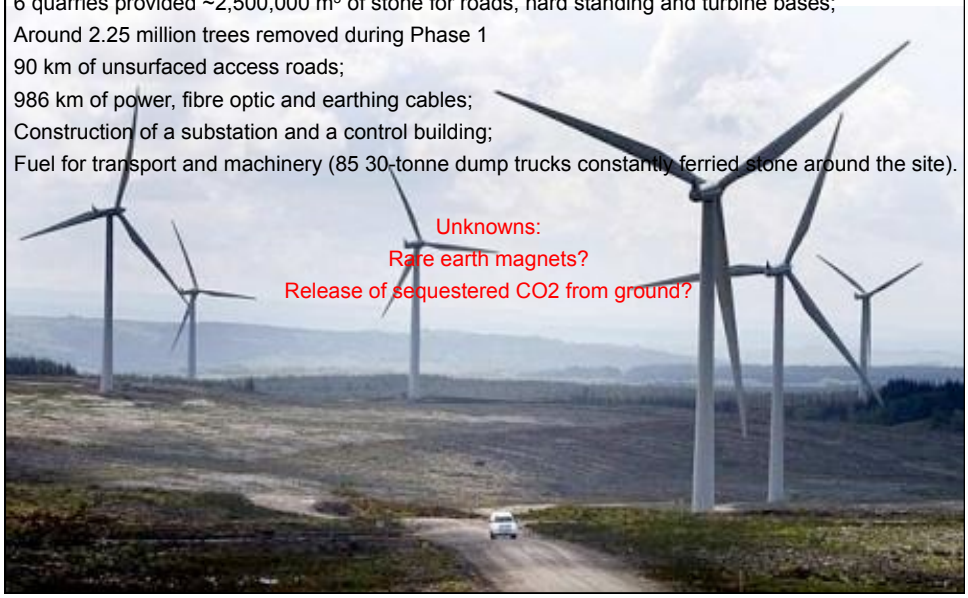


Is this saving the planet?

Whitelees Wind Farm, near East Kilbride, Scotland

Rated capacity 322 MW Load factor ~25%

Each (3MW) turbine base used 500 m3 of concrete, 50 t of structural steel, 2200-3500 m3 of stone;
6 quarries provided ~2,500,000 m^3 of stone for roads, hard standing and turbine bases;
Around 2.25 million trees removed during Phase 1
90 km of unsurfaced access roads;
986 km of power, fibre optic and earthing cables;
Construction of a substation and a control building;
Fuel for transport and machinery (85 30-tonne dump trucks constantly ferried stone around the site).



Unknowns:
Rare earth magnets?
Release of sequestered CO2 from ground?

The subsidies and incentives are part of the system

- EU subsidy regime is not for CO2 reduction but for **renewable energy generation**.
- The two are not equivalent.

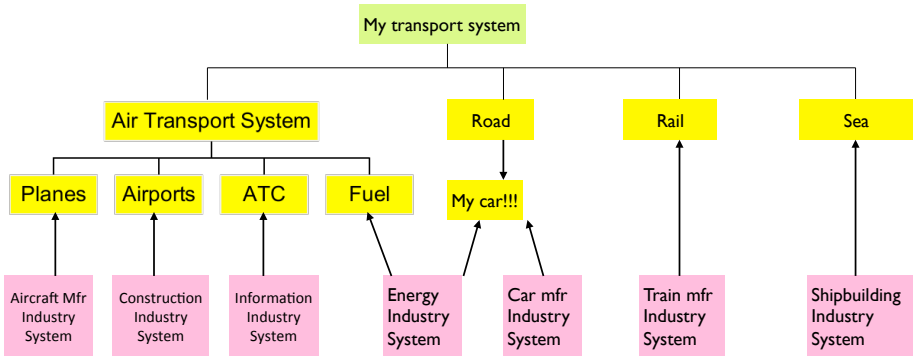
- Is it:
  - economically efficient in reducing CO2 emissions?
  - effective in developing strong and efficient EU renewables industry?
  - accelerating technology innovation? (or just installing mature technology at scale?)
  - strategically effective in reducing dependence on gas imports?

- How to balance subsidy and tax to drive
  - resource efficiency?
  - energy efficiency?
  - industrial competitiveness?
  - environmental stewardship?

SILLITTO ENTERPRISES
ENGINEERING SUCCESS IN A COMPLEX WORLD

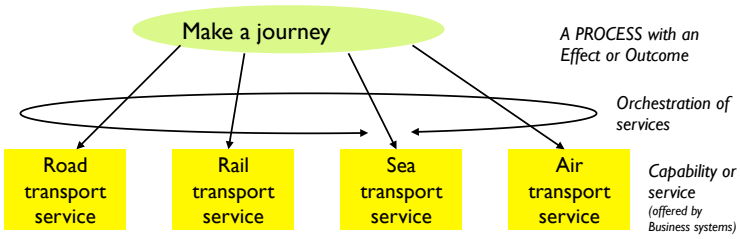
Problems with structuring complex systems in terms of physical hierarchy:

Traditional systems engineering (TSE) advocates hierarchy – usually more or less physical - as the primary structuring tool
This is OK for "closed" systems with a common purpose and budget;
but not for "open" systems in a dynamic market economy.

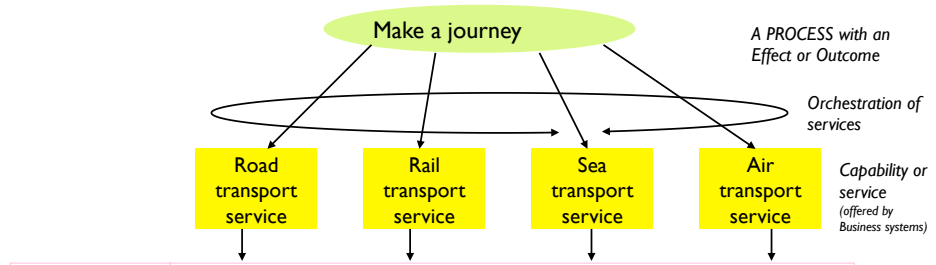


Where does the boss of VW sit in my system hierarchy?
Where does he think I sit in his? How do we make my hierarchy fit his, and yours?

Making sense of complex relationships:
Use case, Process and Service abstractions



Making sense of complex relationships:
Use case, Process and Service abstractions; Components of Capability



Equipment	Car, bus	Train	Ship	Plane
Infrastructure	Roads	Tracks	Ports	Airports
Information	- - - - - Traffic control and co-ord, timetabling, ticketing, - - - - - - - - - - - - -			
Logistics	- Fuel, servicing, spare parts, - - - - - - - - - - - - - - - - -			
"Doctrine"	- Business processes, - - - - - - - - - - - - - - - - - -			
People	Y	Y	Y	Y
Training	Y	Y	Y	Y
Organisation	Y	Y	Y	Y
Finance	Y	Y	Y	Y
Industrial capacity	Y	Y	Y	Y
Science	Y	Y	Y	Y
Technology	Y	Y	Y	Y

Enabling components of Capability or service