Design principles for Ultra-Large-Scale (ULS) Systems

Mr Hillary G Sillitto
Thales UK, 1 Linthouse Road, Glasgow G51 4BZ, Scotland, UK

Abstract: Ultra-Large-scale Systems (ULSS)[1] are a major new challenge for systems and software engineering. Current engineering practice is ahead of the science – we are building systems we do not know how to characterise or analyse, and whose behaviour we cannot fully predict. ULS are characterised by complexity, dominated by emergence, and exist in a state of constant reconfiguration and evolution; all of which make untenable a reductionist approach to engineering and a "closed system" approach to specification and certification.

This paper recommends ten design principles and five design practices for ULS systems, drawing on known systems engineering practice and an understanding of how complexity science is applied in other domains. The paper offers practitioners a strategy and a practical approach to deal with ULS systems – or indeed any system that is larger scale and more complex than those they are accustomed to dealing with – and shows academics some possible routes to addressing the research challenges set out in the SEI report on ULS systems.

Introduction

A ULSS is a system of systems whose scale, rate of evolution and diversity of stakeholders make it impossible for it to be managed as a top down "directed" or "managed" SoS [Rechtin & Maier, 1999]; yet whose intended use requires high levels of trust, safety, security, quality of service, information assurance and data integrity.

Ultra-large-scale (ULS) systems will be interdependent webs of software-intensive systems, people, policies, cultures, and economics. The key literature reference for ULS is [Northrop et al]. It identifies a number of characteristics of ULSS which its authors believe set new and unprecedented challenges for the engineering community in the engineering of ultra-large scale information intensive systems of systems.

Much of the thinking presented in this paper was crystallised by the Socio Technical Systems Engineering workshop held in St Andrews in September 2009 as part of the UK EPSRC funded LSCITS (Large Scale Complex IT Systems) research programme. The paper provides an independent perspective and is not a formal research output.

How this paper is organised

In its major sections, this paper

- Outlines the content and summarises the major conclusions
- Discusses the core issues of ULSS, identifies some key literature references,
- Proposes and discusses five dominant issues for ULSS designers
- Identifies analogies in other domains where similar problems are encountered,

[1] The SEI Ultra Large Sale Systems report: Northrop et al, - -

from which the ULSS community might usefully learn.

- Summarises an understanding of the problem to be solved.
- Sets out the ten proposed design principles
- Sets out the five recommended design practices.
- Summarises issues for researchers and heuristics for practitioners.

Principal Conclusions

This paper concludes that:

1. "Complex systems evolve to the edge of chaos[2]" – and stay there, in a dynamic equilibrium maintained by goal-seeking behaviour. The concept I have chosen to call "stability margin" is a measure of how close a system is to collapse or a dangerous phase transition. Finding pragmatic and relevant measures of stability margin is the key to providing feedback mechanisms to avoid system collapse and maintain system performance, and is a key research challenge.
2. Correct use of Architecture Frameworks (MODAF, DoDAF, NAF) allows practitioners to identify and manage many of the challenging ULS issues.
3. Architecture Frameworks support an open systems perspective. Model driven architecture (MDA) assumes a closed system boundary. So there is a gap to be managed between AFs and Model Driven Architecture implementations.
4. An orthogonal set of issues on policies, values, behaviours and cultures, and emergence due to both socio-technical issues and non-linear system dynamics, are not well addressed either by MDA or Architecture Frameworks.
5. The granularity of the system "chunks" and the extent to which their behaviour is centrally defined and controlled are the key architectural decisions that influence the feasibility of certification and the ability to evolve.

Core issues of ULSS

Linda Northrop presented an overview of the ULSS report to a Socio Technical Systems Engineering workshop held in St Andrews in September 2009 as part of the EPSRC funded LSCITS research programme. This resulted in a wide-ranging and useful discussion among experts and researchers with experience in academia, government and industry. The discussion was held under Chatham House rules so no comments can be attributed to individuals.

Several key conclusions emerged from the debate.

1. Practice is ahead of the science. We are building – or, as we agreed, "composing" is a better word to use in this context - systems that science does not know how to describe or analyse; or if it does, the science is done in domains that the computer scientists studying information-intensive ULS have yet to engage with.
2. The problem is inter-disciplinary and multi-disciplinary. The industry practitioners present believed that the academics interested in ULS, who were predominantly computer scientists, needed to bring in concepts from diverse fields

[2] A phrase often attributed to Stuart Kaufman in connection with his work at the Santa Fe institute

including complexity science, physics, economics and politics[3].

3. There is a wide range of practice in industry between best and worst practice and it is difficult for academics to get a clear view of "best practice" (and therefore the gap between best practice and what is required for success in a given domain) because the best practice is often not visible in the public domain. This makes it difficult to construct a clear research agenda. Is the purpose to move the state of the art forward, or move more practitioners to the state of the art, or both?
4. Current certification techniques and approaches do not transfer readily or at all to open systems.

ULSS will be in a constant state of dynamic adaptation, re-configuration and evolution. ULSS designers need to deal with and accept this inherent variability. [Hollnagel's] system dependability model assumes that there is a continuous multi-dimensional state-space in which we can define acceptable and unacceptable regions of behaviour. We can then map safe regions, risky regions and disaster/catastrophe regions. This suggests design strategies to maintain system stability by detecting and reversing transitions into boundary regions before an unsafe condition is created.

Key issues for ULS will include dependability for safety related functions, and information assurance – will people manage their data and trust other peoples' data? - how will information be protected and shared between permitted users and not be compromised by local failure/drop-outs?

There was a general view that "traditional" Systems and Software Engineering methods are not sufficient and may not even be appropriate for ULS.

Comment: *Traditional methods of structured requirement analysis and design encourage closed system thinking and may lead to a focus on the parts of a system rather than on the interactions between them; whereas at bigger scales it is the emergent properties of the system that are of most interest and concern, and these are created not by the parts themselves but by interactions between the parts.*

Correct application of modern systems architecting techniques and practice mitigates this by focusing on behaviour, of the whole system and at internal interfaces, and of all parts of the system not just hardware and software. But for this to work, the technical and process components of the system need to be well characterised and match the models representing them in all relevant respects. So large systems must be composed of dependable components; feedback needs to be designed in to accommodate known and unavoidable causes of variability (human behaviour, variable quality of service, - -); and "a component good enough for a small system may not be good enough for a large system".

The comprehensive use of modern systems architecting approaches and techniques, properly integrated with engineering governance and delivery processes, and correctly linked between levels and domains, is difficult and is still rare.

It is worth exploring whether wider dissemination and better use of current "best practice" is sufficient to "solve" the ULS "problem".

[3] To the present author, the characteristics of a ULS set out in Northrop's opening presentation appeared to match in close detail the characteristics of large Government acquisition organisations – notably defence – in the Western democracies – see [Gray, 2009].

Dominant issues for ULSS designers

Reflecting on the workshop, **the five dominant issues for ULS** appear to be:

- complexity,
- emergence,
- the people in the system
- constant reconfiguration and evolution
- the organisations that procure, develop and manage the systems

I shall explore these in turn.

Complexity

It is useful to think in terms of two kinds of complexity, SUBJECTIVE and OBJECTIVE [Sillitto, 2009]. Subjective complexity exists in peoples' minds and can be mitigated by consistent clear communication and good stakeholder engagement. Objective complexity is a real attribute of complex systems and is a measure of the extent to which future states of the system cannot be predicted with certainty and precision however good our knowledge of current state and history.

Objective complexity in complex systems means for example that:

- Complex system behaviour is non-linear and non-deterministic; we can't predict future state based on current state or history.
- Complex systems evolve "to the edge of chaos". They stay there in dynamic equilibrium as a result of feedback mechanisms that influence micro-behaviour to maintain macro-stability.
- Complex systems exhibit abrupt "phase changes", which may be reversible, usually with a degree of hysteresis (you have to wind back well beyond the condition just before the phase change to make the system revert to its previous phase), or irreversible, such as an earthquake, bankruptcy, or plastic deformation in a material.
- The system has too many trillions of trillions of states[4] for the behaviour of the whole to be simply deduced from the behaviour of the parts. [Augustine]
- Complex systems exhibit **emergence**, which is mainly a function of **interactions** and **relationships** between the elements rather than of the elements themselves [Hybertson]. Interactions dominate at the technical, and relationships at the social, ends of the socio-technical spectrum [Martin].
- Complex systems are MULTI-SCALE systems. They exhibit short range and long-range order, and micro-scale and macro-scale behaviour. [Sheard and Mostashari]
- Interactions between elements in a complex system can have the same characteristics in different domains. So if we can separate the micro from the

[4] A system with 2 nodes has two one-way communication channels so 4 possible states (both on, both off, either one on and the other off). A system with 6 nodes has 30 available channels (n[n-1]/2). Norman Augustine pointed out in his plenary at INCOSE's 2009 International Symposium in Singapore that this results in at least 2^{30}, or one billion, possible system states for a 6-node system.

macro scale phenomena, we can explain the macro scale phenomena by analogy with other domains where they are well understood.

- Conversely, complex systems may exhibit self-similarity at different scales. This means that patterns – of behaviour and structure – may repeat at different scales. This provides a potential strategy for understanding and managing system behaviour and system design at multiple scales. Fractal self-similarity allows some macro properties to be observed and understood at micro scale.

[Sheard and Mostashari, 2009] describe the implications of these points for systems engineers and the need to develop the skills of "complex systems engineering".

Complexity is difficult to model mathematically. It exists between two other domains that are easier to understand and model, and a third that is not.

- The first is the "linear/random" domain, in which outputs are proportional to inputs and fluctuations are described by random (Gaussian or Bell-curve) statistical distributions.
- The second is the domain of "coherence", exemplified by laser light, conveniently described as "everyone marching in step" – we were all told as school-children that soldiers are ordered to break step when crossing a bridge so that they do not drive it into oscillation and possibly cause it to fall apart. Laser mirrors create feedback that makes the photons "march in step".
- The third is the domain of chaos theory, where bifurcation creates multiple possible outcomes from indistinguishable initial conditions.
- Complex systems are different again. They exhibit what the optical physicist would call "partial coherence" [Born and Wolff], where we don't see the resonant "marching in step", but there is continued correlation of events even through changes to the system.

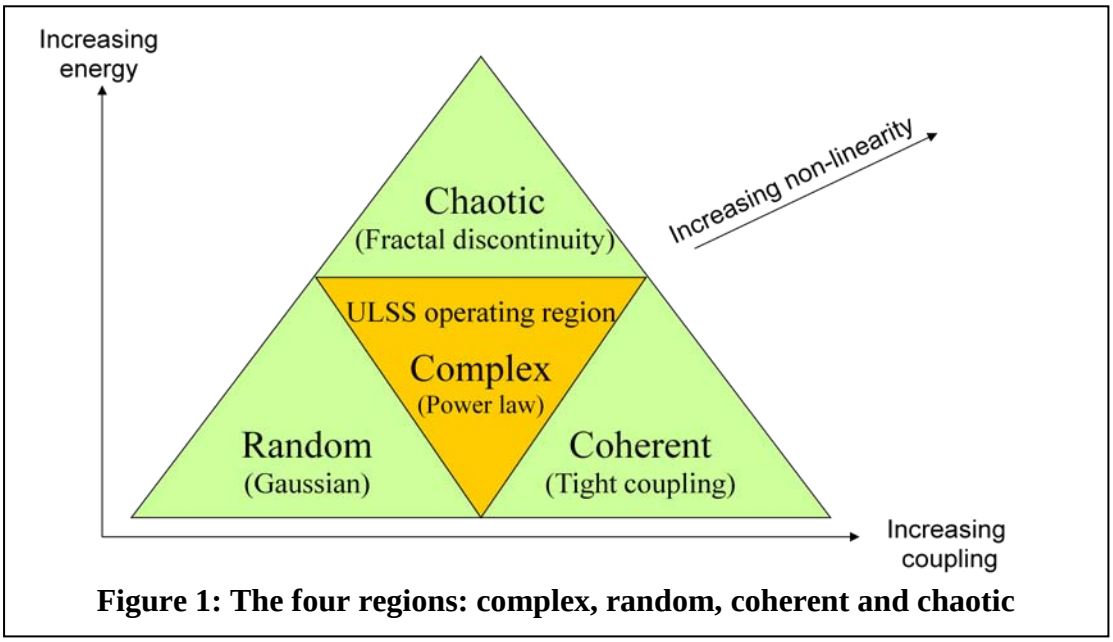

Figure 1: The four regions: complex, random, coherent and chaotic

Complex, coherent and chaotic behaviours are all driven by non-linearity. Non-linearity in the system creates coupling between parts of the system that are not obviously connected, creating correlated outputs from uncorrelated inputs. In real

complex systems on a human and laboratory scale, complexity is often created by resource contention between otherwise unrelated activities. The best-known example of this in the ULSS domain is queuing theory, where resource contention causes increasing delays as the communication channel approaches its limiting capacity. The same behaviour can be observed in motorway traffic, school dinner queues, system development projects, airport baggage handling systems, and global climate change – anywhere different activities are trying to use the same resource at the same time.

The statistics of chance events in a complex system are often characterised by a Power Law distribution, the "signature of complexity" [Sheard 2005]. There are two important things about the Power Law distribution. It is found in a very wide variety of natural and man-made phenomena (earthquakes, risk occurrence in projects [confirmed by Allan, 2010], and possibly in terrorist incidents, e-mail attachment sizes and stock market fluctuations); so once we find it we know we can draw analogies and extract heuristics from other domains where it also occurs [Malamud]. And it means that the probability of a low probability large impact event is much higher than Gaussian statistics would suggest. This has a major impact on ULSS in terms of the occurrence, impact and public acceptance of risk and failure.

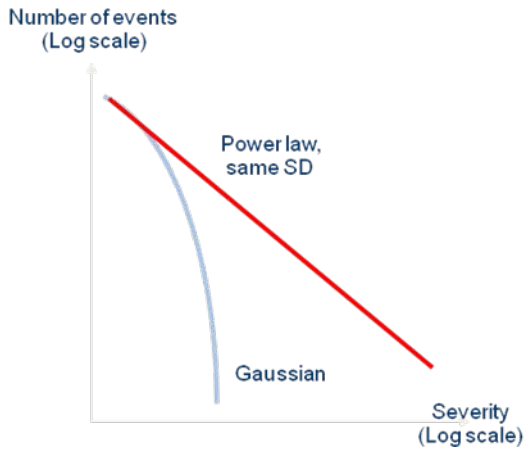

Figure 2: Power Law distribution, the "signature of complexity".
Earthquake statistics and many other natural complex systems follow a power law. Very large impact events are infrequent, but occur much more often than you might expect

Modern astronomy exploits complexity in the interference of light waves, using a phenomenon known as "partial coherence" to estimate the size of distant stars and find un-resolvable double stars. The partial coherence can have a value anywhere between 0 and 1. A value of 0 corresponds to purely random, uncorrelated light such as from a nearby tungsten light bulb. A value of 1 corresponds to fully coherent light, such as from a laser operating on a single frequency with perfect mode quality (referred to as temporally coherent), or from a distant star (referred to as spatially coherent). [Born and Wolff] provide an excellent though daunting mathematical treatment of these phenomena. The analogue of partial coherence in a ULSS might provide a useful stability indicator or system health metric.

Analogues of thermodynamic quantities are also useful. [Derek Hitchins] and [Cloutier] propose Entropy as a macroscopic thermodynamic measure relevant to complex systems. Thermodynamic analogies are useful because we know how to relate microscopic to macroscopic properties, both by measurement, and analytically using the techniques of Statistical Mechanics. Recently, the science has been extended from its roots in classical equilibrium systems to deal with non-equilibrium situations.

A warning is called for however before we enthusiastically adopt mathematical models from established domains. Most mathematical treatments of complex systems assume that the system is made up of large numbers of identical elements with identical behaviours, governed by simple rules – the components are homogeneous,

with low variety (all the same design) and low variability (all built the same). Real ultra-large scale systems are made of elements with high variety (they are different by design) and high variability (their characteristics are not well controlled and may vary over time). So the conclusions from the maths may not transfer, or only give us partial insights.

Emergence

Emergence is the existence of system properties that are caused by the interactions and relationships between elements rather than by the elements themselves.

All systems have emergent properties - ULSS are different only in degree. Unforeseen emergence causes nasty shocks. Many believe that the main job of the systems engineer is to prevent undesired emergent behaviour – to minimise risk. But good systems design also involves maximising opportunity: exploiting emergence to create the required system level characteristics from synergistic interactions between the components, not just from the components themselves. This approach makes the system simpler and cheaper, and, with fewer spurious modes due to unnecessary interactions between components, more predictable and potentially safer.

Many kinds of high-energy systems – for example nuclear power plants - are designed to be fail-safe, so that if the system is perturbed from its intended operating configuration the system will de-tune so that it stops producing power; and if the perturbation gets worse the system will just stop, in a safe low energy state. So the rated power output is not a property of the individual components, but is an emergent property of a correctly tuned system. In a system design that is not inherently safe, detuning could make the power increase to a critical level and cause an accident.

In some cases fail safe is only acceptable in extremis, as it may cause significant loss of life, where fail operational to a much degraded capability is much preferred. This is not unlike nature, with a 'defence in depth' approach to responding to failure (e.g. vegetation response to diverse and increasingly injurious (unpredictable) events, cuts, grazing damage, storms, droughts, infestations, viral attacks, bushfire - generally by diverse 'step-back' and 'recovery' mechanisms). [Jobbins], also see [Jackson]

Some engineering domains are all about creating useful properties through emergence by combining apparently unpromising material or components. In Optics, the individual components of a lens are incapable of producing a good image. Only when they are shaped and combined in very specific ways, to very precise tolerances, can lumps of glass produce high quality images.

So emergence is good, and is already fully exploited and understood in many domains of engineering – and politics, and economics. However, engineering design domains that exploit emergence have good mathematical models of the domain, and rigorously control variability of components and subsystems, and of process, in both design and operation. In optical design we have precise mathematical models of how multi-lens systems behave, well-developed numerical optimisation techniques that allow the design to be driven to converge on a configuration with the desired image forming properties, and manufacturing processes that are very well characterised and are integrated with the design process. The optical industry aims to "design only what we can manufacture, manufacture only what we can model".

The people in the system

In ULSS, people are inside the system. People do not behave in accordance with a mathematical model; nor can they be depended on to do exactly the same thing every time they find themselves in exactly the same circumstances. (But neither, actually, can a complex technical system.) If the system designer cannot RELY on the humans in his system behaving predictably, perhaps he can INFLUENCE them to behave in a way aligned to the goals the system is being designed to support?

Sometimes people are employed to operate systems. Sometimes systems are deployed to help people do their jobs. The distinction is not always clear, particularly to acquirers and designers.

A key challenge in ULSS design is to use the people in the system as part of the solution, not view them as part of the problem. This suggests that ULS systems need to include feedback mechanisms to influence the micro-behaviour by the human actors in the system in such a way as to maintain the macro-stability of the system. This feedback needs to be informed by indicators that can tell whether the system is in a safe operating region or is approaching an undesirable phase transition; and as in any control system, the effectiveness of the feedback mechanism depends critically on the time-constants in the system. (The faster time-scale on which stock market transactions occur nowadays must affect the stability of the world financial system.)

Evidence from other domains suggests that this is an achievable goal. The discipline of quantitative economics popularised by [Levitt] is about relating quantitative measurements and common patterns to the range of individual behaviours. Economists are now able to look beyond the simplistic assumptions of rational behaviour, perfect information and uncorrelated risk to a more complex and subtle understanding of human behaviours in markets. People often have higher priorities than their economic self-interest; market movements create coupling of apparently independent variables through "sentiment" and herd behaviour; and "average" behaviour can be guided through market mechanisms and other incentives.

On the other hand there is much evidence that most people most of the time prioritise short term over long term and local over global interests; and that many will seek to profit from any "market distortions", such as regulatory efforts to maintain stability margins that individuals perceive to be excessive or against their own interests. ULSS designs that take account of this behaviour are likely to be more successful than those that depend on people acting against their own self-interest or intuition.

Collaborative vs competitive behaviours: *Competitive behaviour is the default behaviour of most people most of the time. Collaborative behaviour is essential to allow ULSS to be conceived, deployed and configured. ULSS design needs to provide sufficient benefits in all lifecycle phases to all stakeholders that all will have more to gain by participation in the ULSS than by competitive behaviour that could destroy it.*

Politicians are people too. They may be "outside the system", but they influence the specification of and funding for ULSS. [Rechtin and Maier] devote a chapter to political considerations in systems architecting. While they describe the US, other nations are similar. All large scale systems need a political constituency, which will be sufficiently diverse that different stakeholders will have different priorities and different agendas. So while politicians may be outside the "responsibility boundary" of the system as viewed by the designer, they are certainly inside the "analysis

boundary" that needs to be considered by project sponsors and system architects. They may also have a large say in how the capability provided by a ULSS is used.

Constant evolution and reconfiguration

[Sheard and Mostashari] emphasise the evolutionary nature of complex systems and the need for responsive configuration management and change management.

This evolution and dynamic reconfiguration means that old methods based on fixed system boundaries and frozen requirements simply will not work. Decisions need to be made at the "last responsible moment" [Blockley and Godfrey]. The set of decision timelines largely defines the systems engineering process logic. We must architect – and contract - to accommodate change rather than prevent it. A good approach would be to compose ULSS from components that are stable, dependable, fully characterised building blocks, and to determine system behaviour by managing the interactions between components rather than adjusting the components themselves.

Smart grids will accommodate and manage second-by-second fluctuations in power supply and demand to allow efficient exploitation of renewable energy sources. These are usually distributed and small-scale sources of supply, with inherent fluctuations in individual and local output as a function of the available wind, sun, wave and tide. It is suggested that dynamic pricing will motivate people to make best use of power when it is available. [Mackay] suggests that electric car batteries will be used as temporary storage to balance out short term fluctuations in supply and demand. This is a fundamental paradigm reversal from the assumptions underpinning most current electrical power systems, with their heavy reliance on base-load from fossil or nuclear fuel, and a focus on adjusting supply in order to meet fluctuating demand.

Organisations that procure, develop and manage the systems

Uncertainty and change are part of the territory; current acquisition practices are poorly adapted to this reality. The organisations that procure, develop and manage systems are a major source of uncertainty [Sommerville]. Acquisition organisations themselves show the characteristics of ULSS. Different functional groups within acquisition organisations have different pressures, different incentives, different time horizons, different worldviews, different values and different aspirations.

[Gray] describes how in one such organisation (UK MOD), the incentives and pressures on individual people and teams encourage pursuit of local and single-service interests, rather than global optimisation, and incentivise behaviour that works against the stated purpose of the organisation as a whole. Gray recommends a set of quite radical changes in structure and governance based on his findings.

According to [Gray], *"the Ministry of Defence has a substantially overheated equipment programme, with too many types of equipment being ordered for too large a range of tasks at too high a specification. - - This overheating arises from a mixture of incentives within the Ministry of Defence. In particular, the Armed Forces, competing for scarce funding, quite naturally seek to secure the largest share of resources for their own needs, and have a systematic incentive to underestimate the likely cost of equipment. - - Sensible processes such as "spiral development" and "technology insertion" are heavily discouraged by the current overheated programme - - "bid High Spec, Bid Full Spec", seems to be the encouraged behaviour, however much technical risk that this imposes."*

As noted elsewhere, the organisation is already making a number of changes to rebalance the governance regime to improve both responsiveness to emerging operational needs and coherence across the portfolio of system development projects.

Architectural partitioning in ULSS needs to respect the needs of all stakeholder groups otherwise the system's purpose will be negated by its failure to meet essential requirements of certain stakeholder groups. Programme managers will want to lock down a tight delivery programme. Commercial may want to acquire every part of the system through competition, and/or maintain two or more viable competing supply chains. System architects will prioritise test, integration, and system integrity during operation. Operators demand agility to respond to operational crises. An architecture optimised against any one of these criteria is unlikely to satisfy the others.

Analogies in other domains: the opportunity for ULSS

Hypothesis – we can derive valid and sensible design principles for ULS systems by drawing on analogies and lessons from other complexity domains.

The opportunity in ULSS is to learn from other disciplines that have already encountered these problems. There are many useful analogies.

Town planning and city-sized systems - cities are in constant state of reconfiguration, with areas being rebuilt, infrastructure being replaced, houses being built. [Healey et al] show that cities are immensely complex adaptive systems with most or all of the characteristics of ULS systems. Singapore is an example of a systems engineered society [Liu, 2009]. The City State has been a viable and effective construct for a large part of recorded history, indicating that it is represents some sort of stable optimum form of organization (possibly the largest in which effective top-down governance is feasible [Sommerville]). The UK MOD has recently brigaded its "integrated project teams" into a small number of cohesive and loosely coupled "Operating Centres" to improve the management of its acquisition portfolio. These different perspectives all indicate that there is a critical size of large system up to which certain management styles are effective and above which they are not, or cannot operate with enough nuance and effectiveness to be efficient.

Use of Education and market signals to avoid undesirable behaviour in crisis situations: Singapore has a strong ethos of collective responsibility, reminiscent of the collective ethos that pervaded much of British society in the 50s and 60s. Military mission command depends heavily on training and common culture to make sure that people will act consistently in a crisis and be able to second-guess what their colleagues will do. Both methods lead to self-synchronisation.

Singapore's dynamic road pricing increases the cost of using a road as traffic density rises, so that people are incentivised not to travel at rush hour if they don't need to. Dynamic energy pricing will be key to the effectiveness of smart grids.

Military mission command doctrine trains junior leaders to act on their own initiative in accordance with the commander's intent if communications break down or they find themselves in a situation not anticipated by their orders. This approach is credited [for example by Bungay] for the outstanding effectiveness of the German Army in all theatres at all stages of WW2. This analogy leads us to the concept that in ULS system design each node needs to understand "commander's intent" and have sufficient information and understanding to be able to make local decisions that will result in broadly aligned synergistic behaviour.

Deduction: *Enduring complex open systems are dynamically stable, often "at the edge of chaos", and exhibit goal-seeking behaviour to stay stable in the presence of perturbations. This system level stability is an emergent property of numerous local decisions and actions that have the aggregate effect of maintaining the system in a safe operating region. In such successful open systems the information available to individual decision-makers influences their behaviour in such a way that the aggregate effect of their self-interested decisions is to keep the overall system in, or return it to, a safe operating region.*

The Internet is governed as a voluntary SoS. The Governance regime seeks to provide infrastructure and "joining rules" that enable emergence that creates value for participants. It does not seek to determine all aspects of system and business model in advance. This model sacrifices system integrity to foster unplanned innovation. So a key issue for ULS systems is how to provide high integrity services while simultaneously creating opportunities for innovation on an "open" network.

Understanding the problem situation

There are three fundamental requirements for an effective ULSS.

1. Actors within the system are incentivised to participate in the ULSS because so doing helps them to achieve their own goals.
2. The ULS system has emergent properties that provide enough benefit to the sponsors and critical stakeholders to justify
    a. the additional costs of composing and governing the ULSS and
    b. the risks created by the possibility of unintended emergence.
3. All stakeholders and actors at all scales must have sufficient incentive to behave in a way that maintains ULS system stability and effectiveness over time and ensures recovery after a drop-out.

In addition a number of key issues need to be understood and taken into account.

- New systems and new services may trigger purposeful opposition or "immune responses" from existing systems, and/or create exploitable vulnerabilities.
- A ULSS has no single sponsor, acquirer, supplier, or operator, and no single event denoting entering or leaving service.
- ULSS deployment will be incremental; its lifetime may be indefinite.
- The ULSS "design" must cope with inherent variability both of individual nodes/systems and higher scale behaviour.
- The basic requirement for ULSS resilience is that macro behaviour and overall system integrity must be "fault tolerant" with multiple layers of reversion and recovery since the system will be in constant state of reconfiguration and dropout of individual nodes.

The underlying assumption is that there is a requirement to "compose" a ULSS for a purpose from some combination of legacy, modifiable and new systems and components. The "designer" has to choose and conduct interventions that will achieve the ULSS purpose without having full control of requirements, budget or schedule of the individual systems expected to contribute to the ULSS, nor control over their users and operators. This meets the classic definition of a wicked problem [Rittel &

Webber]. This is illustrated in Fig 3, which was used in the presentation of [Sillitto, 2009] but does not appear in the written paper.

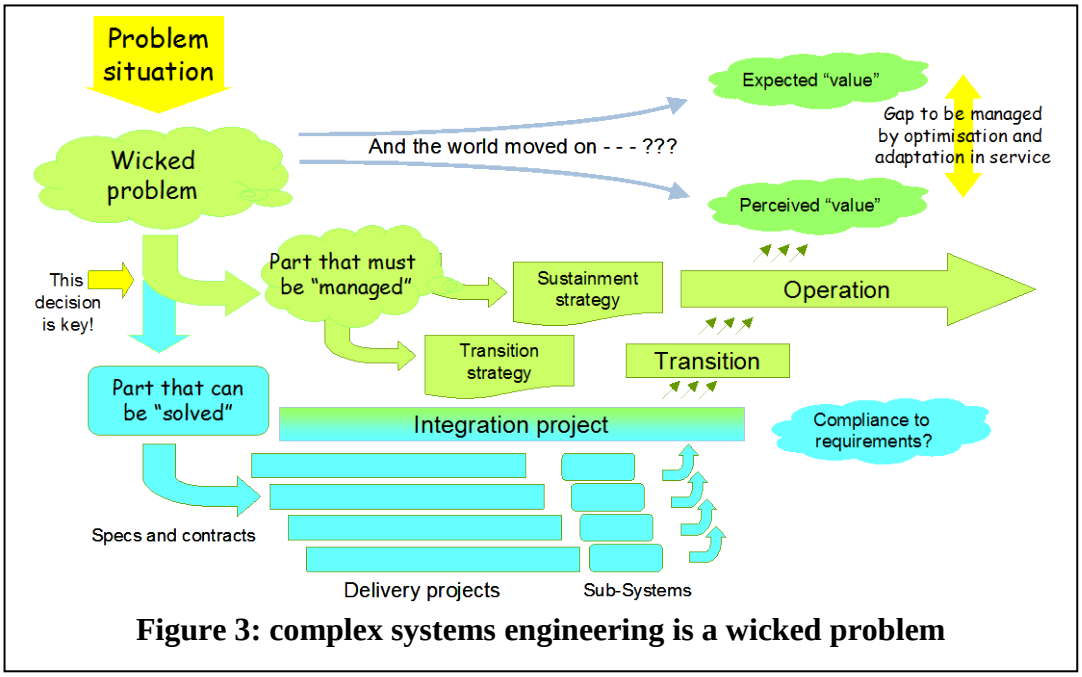

Figure 3: complex systems engineering is a wicked problem

And finally: Do we need a paradigm shift in Government acquisition of software intensive systems? Current government software system procurements are driven by requirements – what sponsors believe they need – whereas commercial ones are driven by architecture – what works, is useful and can be delivered dependably.

Ten ULSS design principles

This section sets out ten proposed design principles for ULS systems. The principles draw on an understanding of how complexity science is applied in other domains and on current systems engineering practice in systems of systems. They are intended to show practitioners a practical approach to deal with ULS systems – or indeed any system that is larger scale and more complex than those they are accustomed to dealing with – and offer academics some possible routes to addressing the challenges set out in the SEI report on ULS systems.

Principle 1: ULSS is a wicked problem. Maintaining system stability while achieving useful emergent behaviour is "a wicked problem that cannot be solved but has to be managed" [Daw]. The ULS system "design" needs recognise this reality, and manage the residual wickedness by identifying appropriate operational rules and system rules, based on Stability Margin, for each node and component of the ULSS. This may be the only degree of freedom left open to a ULS system architect if the intention is to compose a ULS system from legacy systems.

Principle 2: Stability Margin: The fundamental research challenge in ULSS is the problem of identifying and measuring the Stability Margin of a ULSS and using this to provide feedback to operators and system components. This involves extracting real time health indicators with predictive utility, understanding how macro-behaviour emerges from micro-actions and over what time constants, and hence working out how to use the health indicators to provide feedback. The design of the ULSS needs to ensure that the feedback will encourage individual actors to behave in a way that is

consistent with their own interests and goals while also tending to maintain system stability and effectiveness.

Health indicators are the primary criteria for remaining "stable". They are a special case of "real time indicators for satisfaction of needs, wants and expectations". [Jackson] identifies a focus on "capacity" and "the margin heuristic" as key elements in design for resilience.

In principle we could attempt to provide each node in the system with full knowledge, dynamically updated in real time, of the current state of the whole system. This is not a good idea because the state of a distributed system will change faster than it can be measured and notified to a remote node; and because the state space of a complex distributed system is so large that the amount of information we would need to send to each node would swamp the available communication bandwidth. So a surrogate or aggregate or macro-scale measure is essential.

Principle 3: Value adding and Opportunity seeking: System viability depends on risk avoidance or mitigation. But ULSS are composed for a purpose, to create value or to create opportunities for value to emerge. Brian [White]'s paper on Opportunity Management at the enterprise level suggests that ULSS are more often successful when they shift their focus from risk management to opportunity management.

[Rechtin and Maier] say that one of the main things to focus on in architecting is satisfaction of stakeholders. On the other hand, [Rechtin 1991] emphasizes the architect's responsibility for certifiability of the system. Designers and researchers need to focus on purpose and the risk/opportunity balance.

Principle 4: Node and Web architecture: ULSS can be described in terms of a "NODE and WEB" architecture. This is an essential perspective to apply to a ULSS to understand its micro-structure, visualise near-neighbour and local interactions, and analyse "mission threads" in stressing scenarios. Nodes are capable of acting on their own and also of contributing to the ULSS. This contribution should preferably be based on dependable well-defined standardised behaviour – offered service, required service, resource, synchronisation [Sillitto, 2009]. Nodes interact through a WEB of connectivity and relationships, which may be social or technical or both.

The architectural frameworks derived from DODAF (DODAF V2, MODAF V1.1, NAF) - with their concept of "Operational Nodes" and information needlines between nodes – are good for describing node and web architectures. They also provide for behaviour to be specified and controlled by "operational rules" and "system rules" – OV-6b and SV-10b respectively.

Principle 5: Layered Architecture: A layered view of the architecture is needed to allow an understanding of common characteristics of nodes throughout the ULSS, and to provide a consistent level of abstraction that allows a global design approach. [Krob] suggests that "good" layers are the key to managing "the local/global dialectic" and embody the "rules" that allow us to think of the ULSS as "a system".

As an example, Figure 4 shows the "SPIT" model – social, practice[5], information, technology [Sillitto et al, 2006]. mapped to Garstka's "People, Organisation, Process, Technology" (POPT) and to the UK MOD's 8 Defence Lines of Development. These models are different ways of grouping the same basic elements.

[5] This was originally "process". I am indebted to [Martin] for the suggestion of using "practice" both to improve generality and to avoid semantic confusion with "POPT".

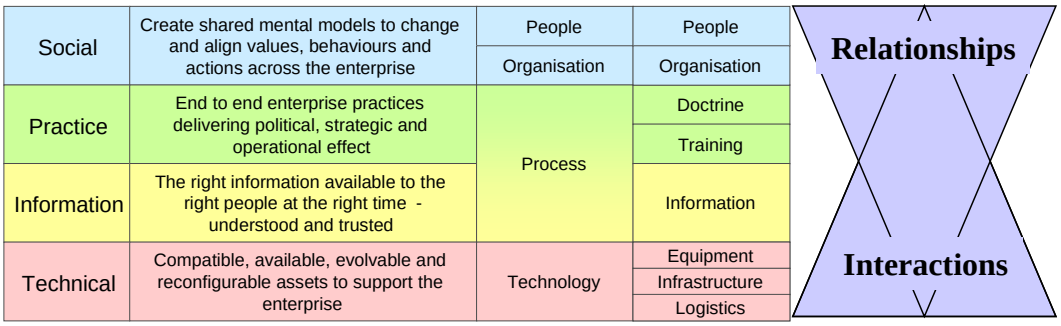

Figure 4; The "SPIT" model.

As we have noted, "relationships" are the principal determinants of emergent behaviour in the upper layers, whereas "interactions" are the principal determinants in the lower layers.

The "SPIT" model shows one type of layering and reinforces [Blockley and Godfrey's] argument that "all hard systems exist inside soft systems". The hard systems (lower layers) deliver performance through interactions; the soft systems (upper layers) create value through relationships. Another valid style of layering is as an explicit or implicit hierarchy, with different layers representing different levels of abstraction or hierarchy. In this style, consistent with the EIA 632 "Building block" model, each layer has to contain all "lines of development" or SPIT components.

Neither a layered view nor a "node and web" view is sufficient on its own to reason about the issues in ULSS design or to specify its elements– both are essential. Most enterprise architecture frameworks have tended to emphasise one of these views at the expense of the other. The introduction of Services views into MODAF may be a sign of a more general convergence between the different architecture "camps".

Principle 6: Align to common purpose at multiple levels: Successful development, deployment, operation, evolution, and retiral/replacement of a ULSS requires stakeholders and operators to adopt synergistic collaborative behaviour throughout the ULS system lifecycle, and at multiple levels in organizational and social hierarchies. To make this happen it is necessary to align Purpose, Policies, Mental models, Incentives, Behaviour and Performance, so that countless individual decisions within the "micro structure" of the ULSS stakeholder and operator community lead to the desired emergent "macro behaviour", which in turn creates the desired benefits.

Principle 7: Understand and manage vulnerabilities: Understand and manage vulnerabilities to failures, to external aggression, and to internal subversion. [Jackson] Operational and system rules need to capture the interaction between operational context, stability margin, operator and system response, and give individual operators guidance that will lead them to act appropriately when the stability margin is degraded by a system vulnerability. Otherwise, for example, as response time slows down, there is an increased likelihood of operators attempting to re-send messages, which further clogs up the system.

Human stress levels and behaviours change according to the pressure of the operational situation and the perceived quality and adequacy of information. [Steve Hitchins] described a situation where command styles in a control room changed radically depending on whether or not the command team felt they had enough

information to understand and assess the situation before they had to make a decision.

Principle 8: Focus on phase transitions: Phase transitions in the ULS system may be irreversible, or deeply disruptive to reverse due to hysteresis, and need particular attention. The ULS design may need to include education and market signals to avoid undesirable coherent behaviour in crisis situations.

Even well-planned phase transitions in well-designed complex systems - going live, introducing a major change, introducing a minor change that has pervasive effects across the whole system - will cause unintended and possibly adverse consequences. Numerous examples include the highly publicized problems on the opening day of Heathrow Terminal 5 [House of Commons Transport Committee]. Risk mitigation strategies must include expectation management in the period before going live; and contingency plans must cover effective, accurate and timely communications with staff, customers, sponsors and the media [Mitroff and Anagnos].

It is important to be able to distinguish between extreme fluctuations within "business as normal" and a more fundamental transition leading to things changing in some irrevocable way - it may take years to tell the difference. The measurement strategy needs to provide leading indicators of approach to dangerous phase transitions. Predictable and planned phase transitions need to be treated as high risk events.

Principle 9: Provide feedback relevant to decision cycles and available choices: The ULS system design needs to include feedback that is relevant to the decision cycles and available choices of individual actors within the system, so as to ensure that the "drift velocity" or aggregate effect of rational self-interested decisions by local actors keeps the macro system stable and aligned to purpose. We may need to keep perturbing the system to keep people aware of the risk of collapse/catastrophe and so behaving in a risk aware manner. Having seen several, I am increasingly persuaded that stock market crashes happen when people have lost risk awareness and can't correctly interpret symptoms that the stability margin has been eroded.

The design also needs to take into account the risk and effect of purposeful opposition. The system will usually be operating "at the edge" where it is most efficient in steady state, and may be vulnerable to "intentional bad behaviour" (such as hijacking, suicide bombing, or insider trading).

Principle 10: Stable intermediate forms: Constant reconfiguration and evolution is a fundamental characteristic of ULS systems. Complex systems evolve most rapidly through "Stable Intermediate Forms" [Rechtin, 1991]. So each capability increment for the ULS as a whole, and the increment's deployment on any individual node, and the change as any individual node joins or leaves the ULSS, should all be stable intermediate forms. The software industry's "daily build and smoke" approach [McConnell] is another example of the application of the stable intermediate forms heuristic – the software system is never more than a day away from a working build.

Five ULS design Practices

The design practices offered below attempt to show how existing known good systems engineering practice can deal with many of the ULSS challenges and implement many of the principles listed above.

Practice 1: Physical architecture: Defining a node in a ULS system: A node typically has 8 components, as follows:

Organisation Process Training People

Data Software Hardware Infrastructure

The bottom four components are elements of a traditional "technical system". Their behaviour can be aggregated as a "service" provided to the "social system" in the upper four elements. Software and hardware are programmed to manage data within an infrastructure context. People are trained to manage information within an organisational and process context. The infrastructure defines the physical web, where behavior is determined predominantly by *interactions*. The organisation defines the social web, where behavior is determined predominantly by *relationships*.

MODAF V1.1 has introduced the concept of the Capability Configuration, a cohesive capability element constructed from a number of types of system element (hardware, software, people, - -). This construct is well adapted to defining the logical structure of nodes. It is however blind to policies, values, and cultures.

The ULSS problem is to define dependable nodes and make the web resilient to nodes joining/leaving/dropping out/malicious behaviour, without any node having full knowledge of the overall system configuration and state at any given time. Required and acceptable behaviour can be defined in MODAF using the Operational and System Rules Models, and should cover both "normal" and "exceptional" conditions.

The granularity of these system nodes or "chunks", and the extent to which their behaviour is centrally defined and controlled, are the key architectural decisions for the ULSS and greatly influence the feasibility of certification.

Practice 2: Functional architecture: Specify behaviour as services offered and requested between nodes and components: There are four kinds of service:

- Autonomous (node provides service to local user with local data)
- Node consumes service from other node
- Node provides service to other node
- Nodes collaborate to create emergent services.

The methods of SOA (Service Oriented Architecture) view the world in terms of low level "information services" exchanged within the technical layer, orchestrated to create "business services" that usefully support people doing business processes. A business service can and should be defined without reference to how it is implemented. Only the function, the performance and integrity levels should be specified, leaving the service provider free to optimise the implementation. "A document can be translated by person or machine, the differences being in terms of time taken, form of output, and ability to cope with nuance and errors" [Davidson].

Service-oriented approaches claim to "place complexity where it is best managed".

By defining interface behaviour in terms of services we avoid the need to understand internal detail of nodes (except for assurance and repair purposes) and can reconfigure services within understood parameters provided the components provide services as specified. In theory we can operate in a network in blissful ignorance of where information is coming from and who else can see it – as people do on Facebook. This will not always be an acceptable state of affairs in high integrity systems.

Many argue that we need a higher level of abstraction than "services" to deal with ULSS. [Somerville] suggests that what we are interested in is "situated functionality" that changes dependent on the environment and the client.

Practice 3: Synchronise effects with Control levers: Desired ULS system emergent properties are created by dynamic behaviour. This dynamic behaviour needs to be influenced and synchronised across the "web" (ULSS are too complex to depend on top-down direction). So a major element of ULSS design is the choice of measures and mechanisms to influence individual nodes and actors to behave synergistically.

Available control levers include training, informing, top-down direction, and "market mechanisms" that incentivise desirable behaviour. It's a control system problem. The right measures and mechanisms will depend on time constants – how quickly information and effects propagate through the system, and decisions are made; and on the gain – the effect produced by each action. If decisions are based on stale information and produce delayed effects, there will be delays and overshoots in response, the system may become unstable, the effect of actions will not be what was intended, and the desired emergent properties will not be achieved.

Practice 4: Understand analysis and responsibility boundaries: There is a clear "responsibility boundary" around the part of the system we are responsible for; but the value-added of our part depends on its interaction with the wider containing system. Many trade-offs and design decisions depend on a good understanding of the wider system. That part of the wider system we need to understand to do these trade-offs is contained within the "analysis boundary". [Sillitto, 2005]

Most systems engineers working in ULS systems will have a limited span of control due to the dispersed management structure. Classical top-down systems engineering is therefore not possible, or at best heavily constrained. They need to understand: the boundary within which they are responsible; the part their system or service plays in the overall capability; and how to manage the dependencies between their component of the system and the other elements of the ULS system it interacts with in order to achieve the required emergent properties and avoid introducing unacceptable ones.

So, where possible, select the system boundaries carefully in a way that allows holistic self-organising behaviour to move things in generally the right rather than generally the wrong direction.

Principle 5: effective and appropriate measurement: We have suggested that emergent value in ULS systems depends on relationships, while performance depends on interactions. Interactions and performance are easy to measure, while measuring relationships and value is difficult. A measurement strategy that focuses only on interactions is unlikely to foster relationships that deliver emergent value.

"What can be measured gets locked into the contract". There is a bias towards measuring what is easily measurable (availability, transaction rates, responsiveness) rather than what supports a relationship that achieves the system's purpose.

"On a >$1Bn managed service deal we measured 144 SLA parameters that operationally defined the service, none of which meant much in relation to the business the service was meant to support, but neither could they be ignored. My analysis showed there was little or no correlation between what the customer really valued and what they were buying." [Yearworth]

Understand what needs to be measured and locked into contracts to drive performance (to achieve system stability) and behaviours (to create potential for added value); and what needs to be measured but not contracted to detect unintended consequences. Lean value stream analysis may allow "value" to be determined and related to lower level measurable interaction and performance metrics.

Uncertainty is inevitable must be accepted and recognized. An approach which has proved successful for this is documented by [Blockley and Godfrey] as the "Italian Flag" method.

Summary and conclusions

In general: We need to get a better handle on the right metrics, "styles" and patterns for systems of systems at different scales, and how they translate between domains.

Issues for practitioners: Architectural frameworks – DODAF, MODAF, NAF – provide a set of tools and constructs adapted to open system problems, and allow us to differentiate and link between operational (social and process) and system (information and technology) perspectives. They are well suited to describing ULSS in a way that allows some the issues discussed in this paper to be identified and analysed. They do not however address policies, cultures, behaviours and values.

There is a gap between Architecture Frameworks (AFs) and Model Driven Architecture (MDA). AFs take an open system view whereas MDA takes a closed system view. We need an open system view to understand the problem and identify the "closed system components of the solution". These components can be implemented and their implementation controlled using MDA. We also need to develop management strategies for the "residual wickedness", and systematically analyse and test for conditions likely to lead to the onset of non-linear behaviour.

The granularity of the system "chunks" and the extent to which their behaviour is centrally and completely defined and controlled are the key architectural decisions that influence whether the ULS can self-optimise and the feasibility of certification.

Issues for researchers: "what can be certified" both in terms of dependable components, and interaction between nodes in a "Web".

How to determine appropriate measures of residual stability margin for different kinds of software intensive systems, and how this information can be measured in real time and used to incentivise collaborative behaviour that will preserve system stability and recover stability margin.

How to deal with issues of Trust, uncertainty, and incentivizing collaborative behavior, relating particularly to understanding of complex systems by users & operators, trust of the system itself, and of the information it presents and coping with the effect of uncertainty on public perception of risk; and how to cope with the inevitable competitiveness and short-termism among some or many stakeholders.

How to balance risk and opportunity in system assurance; and how to integrate "hard" design processes with social integration of policies, values and behaviours.

Acknowledgements

I would like to thank colleagues too numerous to mention in Thales, MOD and INCOSE whose insights in conversation and correspondence have helped to formulate and refine these ideas, and also and in particular Ian Sommerville, John McDermid, Dave Cliff, Brian Collins, Patrick Godfrey, Stuart Jobbins, James Martin, Neil Allan, Daniel Krob, Scott Jackson, Linda Northrop, and the members of the LSCITS National Stakeholder Board. I also thank Thales for permission to publish.

References

Allan, Neil, University of Bath, *private conversation re his current research*, 2010

Augustine, Norman, *Plenary speech*, INCOSE International Symposium 2009

Born, Max, and Wolf, Emil, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light (7th ed.)*, Cambridge University Press (1999) ISBN 0-521-64222-1

Bungay, Steven, *Alamein*, Aurum Press, 2002 ISBN 1 85410 929 4

Blockley, David, and Godfrey, Patrick, *"Doing it differently - systems for rethinking construction"*, ISBN 0727727486, Thomas Telford Limited, 2000

Cloutier, Rob – *Entropy in system architecture* (TBC), INCOSE IS09

Daw, Andrew: *Capability as a wicked problem*, AIAA Conference Belfast 2006(?)

Davidson, Murray – *private conversation*, Oct 2009

Gray, Bernard: *Review of Acquisition for the Secretary of State for Defence - An independent report*, October 2009

Healey, Cameron, Davoudi, Graham and Madani-Pour (editors), *Managing Cities: The New Urban Context*, Wiley, May 1995

Hollnagel, Erik, et al: *Resilience engineering: concepts and precepts*, Ashgate publishing, 2006

Hollnagel, http://erik.hollnagel.googlepages.com/whatisresilienceengineering%3F

Hitchins, Derek: *Advanced Systems Thinking, Engineering and Management*, Norwood MA: Artech House, 2003

Hitchins, Steve: *NITEworks presentation*, EA Conference London Feb 08

House of Commons Transport Committee, 2007–08: *The opening of Heathrow Terminal 5*, Twelfth Report of Session 2007–08

Hybertson, Duane, *Model-oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*, CRC Press 2009, **ISBN** 9781420072518

Jackson, Scott, *Architecting resilient systems – accident avoidance and survival and recovery from disruptions*, Wiley © 2010, **ISBN** 978047040503-1

Jobbins, Stuart – *e-mail correspondence*, 10 Nov 09

Krob, Daniel – *e-mail correspondence* March 2010

Levitt, Steven, and Dubner: *Freakonomics: A Rogue Economist Explores the Hidden Side of Everything*, Wm Morrow, 2005, ISBN 0-06-123400-1 (Hardback), ISBN 0-06-089637-X (large print paperback)

Liu, Pao Cheng: *Large Scale Systems Engineering in a Small Country – the Singapore Story*, Closing plenary at INCOSE IS09 Singapore July 2009; reported by Chua and Chia, INCOSE Insight Vol 12 Issue 3 Oct 2009

Mackay, David, "*Renewable energy – without the hot air*", UIT Cambridge Ltd, 2009

Martin, James: *e-mail correspondence* 20 Dec 2009

McConnell, Steve – *Rapid Development: Taming Wild Software Schedules.* Redmond, Wa. Microsoft Press, 1996. ISBN: 1-55615-900-5.

Malamud*; 'Tails of natural hazards'*, Physics World, Aug 2004

Mitroff, Ian, with Gus Anagnos; *Managing crises before they happen – what every executive manager needs to know about crisis management*, Amacom, 2000

Northrop et al, *Ultra-Large-Scale Systems: The Software Challenge of the Future*, SEI/CMU, ISBN: 0-9786956-0-7, Published: June 2006

Rechtin, Eberhardt, Prentice Hall (1991) – "Systems Architecting – creating and building complex systems"

Rechtin & Maier – The Art of Systems Architecting (1999?)

Rittel, Horst, and Melvin Webber; *"Dilemmas in a General Theory of Planning,"* pp. 155-169, Policy Sciences, Vol. 4, Elsevier Scientific Publishing Company, Inc., Amsterdam, 1973. [Reprinted in N. Cross (ed.), *Developments in Design Methodology*, J. Wiley & Sons, Chichester, 1984, pp. 135-144.]

Sillitto, 2005 – *"Some really useful principles – a new look at the scope and boundaries of systems engineering"*, INCOSE IS05

Sillitto et al, 2006 – *Engineering NEC – Basic Principles*, MOD Integration Authority, © Crown Copyright 2006

Sillitto, 2009 – *On Systems Architects and Systems Architecting: some thoughts on explaining and improving the art and science of systems architecting*, INCOSE IS09

Sheard, Sarah, 2005. Comment while presenting *Practical Applications of Complexity Theory for Systems Engineers,* INCOSE International Symposium Rochester 2005,

Sheard and Mostashari, *Principles of complex systems for systems engineers*, Systems Engineering, Wylie, 2009

Sommerville, Ian, *e-mail correspondence*, 18 Nov 2009

White, Brian: *Enterprise Opportunity and Risk*, INCOSE IS 2006

Yearworth, Mike: *e-mail correspondence* 3 Jan 2010

Biography

Hillary Sillitto has a Physics degree, became an optical engineer, then found his skills useful in other domains and complex projects. He started his engineering career in 1976 and joined Thales in 1993. He was INCOSE UK Chapter president in 2004-2006, Head of UK MOD Integration Authority from 2005 to 2008, and is a member of the LSCITS National Stakeholder Advisory Board. He holds several patents, is a Thales Expert, a Chartered Engineer, a Fellow of the Institute of Physics, an INCOSE fellow, a visiting Professor at the University of Bristol, and has achieved INCOSE certification as an Expert Systems Engineering Professional (ESEP).