The UK Large-Scale Complex IT Systems (LSCITS) Initiative

Radu Calinescu, University of Aston
Dave Cliff[1], University of Bristol
Justin Keen, University of Leeds
Tim Kelly, University of York
Marta Kwiatkowska, University of Oxford
John McDermid, University of York
Richard Paige, University of York
Ian Sommerville, University of St Andrews

December, 2010.

Abstract

This paper summarizes the current activities and the future plans of the UK National Research & Training Initiative in the Science & Engineering of Large-Scale Complex IT Systems (LSCITS). The LSCITS Initiative is funded by over £10m of UK public funds, runs from 2007-2014, and involves research teams at six top British universities, including a training programme that aims to graduate more than 50 doctoral researchers. We start with a discussion of the Initiative's motivating concern: the observation that in most advanced economies, large-scale socio-technical systems are increasing in complexity, and in socio-economic criticality, while our ability to engineer and manage such systems is probably not increasing at the same pace. From this comes the concern that in advanced economies we may already be reliant on large-scale complex IT systems that support critical social and economic functions, and yet for which we cannot predict their failures until it is too late. We then describe the structure of the UK LSCITS Initiative, summarize some current work, and describe our future plans.

1. Introduction

For what events will the date of May 6th, 2010 be remembered? In Britain, there was a general election that day, which ousted the ruling Labour Party after 13 years and led to the formation of the UK's first coalition government since 1945. Nevertheless, it seems likely that in financial circles at least, May 6th will instead long be remembered for dramatic and unprecedented events that took place in on the other side of the Atlantic, in the US capital markets. May 6th is the date of what is now widely known as the "Flash Crash".

On that day, in a period lasting roughly 30 minutes from approx 2:30pm to 3:00pm EST, the US equity markets underwent an extraordinary upheaval: a sudden catastrophic collapse followed by an equally unprecedented meteoric rise. In the space of only a few minutes, the Dow Jones Industrial Average dropped by 660 points, its biggest ever one-day loss of points, representing the disappearance of around one trillion dollars of market value. In the course of this sudden downturn, the share-prices of several blue-chip multinational companies went haywire, with shares in companies that had previously been trading at a few tens of dollars plummeting to $0.01 in some instances, and rocketing to values of $100,000 in others. Seeing prices quoted by some major exchanges suddenly going haywire, other major exchange-operators declared "self-help"

[1] Contact author. Email dc@cs.bris.ac.uk. Phone +44 79 77 55 22 50.

(that is, they invoked a regulation allowing them to no longer treat the price-feeds from the other exchanges as valid), thereby decoupling the trading on multiple venues that had previously been unified by the real-time exchange of reference price data.

Then as suddenly as this downturn occurred, it reversed, and over the course of another few minutes most of the 600-point loss in the Dow was recovered, and share prices returned to levels within a few percentage points of the values they had held before the crash.

While some equity spot and derivatives trades that took place at the height of the mayhem were subsequently "busted" (declared to be invalid on the basis that they were clearly made on the basis of erroneous data) by the exchanges, the means by which trades were selected for busting was argued by many to be arbitrary, after-the-fact rule-making. Some traders who had lost large amounts did not have their trades busted; some who had made handsome profits found their gains taken away. The flash-crash chaos had rippled beyond the equity markets into the foreign exchange (FX) markets where certain currency exchange rates swung wildly on the afternoon of May 6th as the markets attempted to hedge the huge volatility and risk that they were suddenly seeing explode in equities. There is no provision to retrospectively bust trades in FX, and so those deals were left to stand. Sizeable fortunes were made, and sizeable fortunes were lost, by those caught in the storm; the issue of who lost and who gained was almost random.

Two weeks later, the US Securities and Exchange Commission (SEC) and the Commodity Futures Trading Commission (CFTC) jointly released an interim report into the events of May 6th (CFTC&SEC, 2010a) that established little, other than dispelling rumours of the flash crash having been caused by a "fat-finger" error (where a trader mis-keys an order) or terrorist action. After that, for more than four months there was open speculation on the cause of the flash crash, and senior figures in the markets voiced their growing exasperation at the lack of a straightforward explanation. Identifying the cause of the crash was made difficult by the "fragmentation of liquidity" (trading taking place simultaneously on a number of independent but interconnected exchange-venues), the consequent lack of a single unifying "consolidated tape" showing unique timestamps for all events in all the markets, and the widespread use of algorithmic trading systems: autonomous adaptive software systems that automate trading jobs previously performed by human traders, many operating at super-human speeds. Various theories were discussed in the five months that it took the SEC and CFTC to produce their joint final report on the events of May 6th, many speculated on the role of high-frequency trading (HFT) by investment banks and hedge funds, where algorithmic traders buy and sell blocks of financial instruments on very short timescales, sometimes holding a position for a only few seconds or less. When the SEC/CFTC final report on the Flash Crash was finally published on September 30th, nearly five months after the event (CFTC&SEC, 2010b), it stated that the trigger-event for the crash was a single block-sale of $4.1bn worth of futures contracts, executed with uncommon urgency on behalf of a traditional fund-management company. It was argued that the consequences of that trigger event interacting with HFT systems rippled out to cause the system-level failures just described. The SEC/CFTC report was met with very mixed responses. Many readers concluded that it left more questions unanswered than resolved.

We argue here that the Flash Crash is best understood as a failure in a large-scale complex socio-technical system of systems (SoS), and that there are other such socio-economically significant SoS in which similar problems or failures seem likely, or at least plausible, in future. Unpacking that assertion requires some care, so we'll start first with a discussion of notable technology failures, then bring the conversation back to discussion of failures of the financial markets, and then extrapolate out to other large-scale complex socio-technical SoS. Such SoS are very often the result of organic growth and unplanned accretion rather than clean-sheet engineering design, thereby involving or acquiring significant degrees of variability in components and heterogeneity of constituent systems. For this reason traditional engineering techniques cannot necessarily be trusted to deliver acceptable solutions. Therefore, new approaches are required.

2. Background: Failures in Risky Technology

The global financial markets are not the only area in which the application of new technologies has led to failures. Although operator error can be attributed to many failures, as technological systems grow in complexity the prospect of failure-modes being inadvertently designed-in also grows. Take, for example, bridge building. As an engineering activity this is something that dates at least as far back as ancient Rome (c.150BC) and so probably doesn't figure as a risky technology for many people. Yet for decades, engineering students have been taught the story of the Tacoma Narrows suspension bridge, opened in July 1940, which collapsed four months later, where the designers did not anticipate the prospect of wind-flows over the bridge deck reinforcing the deck's natural mode of vibrations, leading to the bridge shaking itself apart. Presumably, current and future students will also be taught the story of the London Millennium Bridge, which opened in June 2000 and two days later was closed for two years to remedy destabilizing swaying motions induced when groups of people walked over it. A significant difference between Tacoma Narrows and London Millennium is that in the latter case, it was the interaction of people, the users, with the engineered system that caused the problem. The Millennium Bridge on its own, as a piece of engineering, was a fine and stable structure; but when we consider the interaction dynamics of the larger system made up of the bridge and its many simultaneous users, there were serious unforeseen problems in those dynamics that only came to light when it was too late.

As engineered systems become more complex, it becomes more reasonable to argue that no one person or group of users was responsible for failures, but rather that the failures were inherent, latent, in the system; this seems especially so in the case of *socio-technical systems*, i.e. systems whose dynamics and behaviour can only be properly understood by including human agents (such as operators and/or users) within the system boundary.

This is perhaps most clear in some of the more famous technology failures of the past 40 years. The oxygen-tank explosion that crippled the *Apollo 13* Lunar Service Module as it was en route to the moon in 1970, and subsequent safe return of her crew, has been rightly popularized as a major triumph of bravery, skill, teamwork, and engineering ingenuity. Nevertheless, the fact remains that NASA very nearly suffered the loss of *Apollo 13* and her crew, due to the compounding effect of several independent small failures of process rather than malign intent or major error from one or more individuals. The successful return of *Apollo 13*'s crew owed an awful lot to the availability of accurate simulation models, physical replicas on the ground of key components of the spacecraft, where recovery procedures could be rehearsed and refined before being relayed to the astronauts. The value of simulation models is something that we will return to later in this paper.

While loss of a space vehicle is undoubtedly a tragedy for those concerned, the number of fatalities is small in comparison to the potential losses in other high-consequence systems, such as petrochemical plants and nuclear power stations. The release of toxic gas at the Union Carbide plant in Bhopal in December 1984 immediately killed over 2,000 people, with estimates of the subsequent delayed fatalities running at 6,000-8,000. The partial meltdown at the Three Mile Island nuclear plant in 1979 was successfully contained, but the reactor-core fire at Chernobyl in 1986 was not, and the number of deaths resulting from that event is widely held to be many thousands.

High-risk technology failures including *Apollo 13* and Three Mile Island were the subject of serious scholarly analysis in Charles Perrow's seminal work *Normal Accidents* (Perrow, 1984). Perrow argued that in tightly-coupled systems with sufficiently complex internal interactions, accidents and failures, including catastrophic disasters of high-risk systems with the potential to end or threaten many lives, are essentially inevitable – such accidents are, in that sense, to be expected as "normal", regardless of whether they are common or rare.

In Perrow's terms, the losses of the NASA space shuttles *Challenger* in January 1986 and *Columbia* in February 2003 were also normal accidents. However, the sociologist Diane Vaughan argued for a more sophisticated analysis in her classic study *The Challenger Launch Decision* (1997), in which she gave a detailed study of transcripts, covering the hours immediately preceding *Challenger*'s launch, of interactions between NASA staff and the staff of Morton Thiokol, manufacturers of the shuttle's solid-fuel rocket booster (SRB) that failed leading to loss of the vehicle and her crew. The transcripts had been released as part of the official Presidential Commission on the Space Shuttle *Challenger* Accident, led by William Rogers. A shocking finding of the investigation was that the specific failure-mode (burn-through of rubber O-ring seals in a critical joint on the SRB) had been known since 1977 and the consequent potential for catastrophic loss of the vehicle had been discussed by NASA and Thiokol, but the shuttle had not been grounded. Vaughan concluded that while the *proximal* cause of disaster was the SRB O-ring failure, the *ultimate* cause was a social process that Vaughan named *normalization of deviance*. Put simply, normalization of deviance occurs when the safe-operating envelope of a complex system is not completely known in advance, and where events that were *a priori* thought to be outside the envelope, but which do not then result in failures, are taken after the fact as evidence that the safe envelope should be extended to include those events. In this way, deviant events become normalized: the absence of a catastrophe thus far is taken as evidence that in future catastrophes are less likely than had previously been thought. The flaw in this line of reasoning is starkly revealed when a catastrophe then ensues. In Vaughan's analysis, the loss of *Challenger* was not a purely technical issue but rather was an organizational failure in the *socio-technical system* comprised of the (technical) shuttle hardware systems and the (social) human individuals, teams, and organizations that had to interact appropriately to ensure safe launch and return of the shuttle.

Vaughan's analysis of the *Challenger* accident came more than a decade after the official inquiry into that 1986 event. In contrast, immediately following the loss of *Columbia* in 2003, because of her work on *Challenger*, Vaughan was invited onto the Columbia Accident Investigation Board (CAIB) and subsequently authored a chapter of the CAIB official report. It was argued that once again an organizational failure at NASA had resulted in loss of a vehicle, via a long-standing process of normalization of deviance.

For *Columbia*, the *proximal* cause of disaster was a lump of insulating foam that broke away from the external fuel tank and struck the leading edge of the orbiter's left wing, damaging its thermal insulation: on re-entry, this damage allowed atmospheric gases, compressed in the bow-wave at the wing edge and hence heated to more than 1,500 Celsius, to penetrate the wing, and the vehicle then broke up at high speed. But the *ultimate* cause was an organizational culture that had once again engaged in normalization of deviance. Prior to loss of *Columbia*, sixty-four previous missions had suffered strikes from insulating material breaking away during launch and hitting the orbiter, each such strike was technically a violation of design requirements (most notably, in 1988 on mission STS-27, insulation broke away from an SRB and damaged 700 of the heat-insulating tiles on shuttle *Atlantis*) but repairing the damage from insulation strikes became increasingly seen as a routine maintenance issue (Mullane, 2006). Vaughan discussed the similarities between the *Challenger* and *Columbia* losses in a book chapter (Vaughan, 2005) and has documented her experience on the CAIB and her subsequent interactions with NASA in a 40-page journal article (Vaughan, 2006). The CAIB report is probably the first major US government accident investigation that explicitly states the cause of the disaster to be a socio-technical system failure.

The approaches exemplified by the writings of Perrow and Vaughan are not the only ones. Studies of so-called High-Reliability Organizations (HROs) such as emergency rooms in hospitals, firefighter teams, and the flight-deck operations crews on aircraft carriers have revealed that there are social and organizational, as well as technical, solutions to creating resilient socio-technical systems: see, for example, Roberts (1990); Weick & Sutcliffe (2007); and Reason (2008).

But what does this academic literature on the study of technology failures offer to teach us about the events of May 6th, 2010? Of course, the Flash Crash was by no means the first failure in a major financial market. As anyone reading this paper must surely be aware, in July 2007 the investment bank Bear Stearns was the first in what turned out to be a sequence of major financial institutions to signal that it had suffered significant losses on subprime hedge funds, triggering a sudden dramatic reassessment of counterparty risk in most major financial institutions around the world which led, *inter alia*, to the UK's Northern Rock consumer bank being the first to suffer a full-scale bank run in 150 years; and to the US government bailing out insurance giant AIG, mortgage providers Freddie Mac and Fannie Mae, and yet famously not extending a lifeline to Lehman Brothers, which turned out not to be too big to fail, and which duly went bust.

Taking a longer historical perspective, the crisis of 2007-08 was just one in a sequence that stretches back through the collapse of the LTCM hedge-fund in 1998; the "black Monday" crash of October 1987; the US savings-and-loan crisis of the mid-1980's; the Wall Street Crash of October 1929; the South-Sea Bubble of the 1720s; and the Tulip Mania of the 1630s.

This history of financial crises has been documented in a popular text by Kindleberger (2001), and the events of 2007-08 have been recounted from a number of journalistic perspectives, of which Lewis's (2009) and Tett's (2009) are notably thorough and well written. Tett's perspective is particularly insightful: she is a senior journalist for the *Financial Times* but has a PhD in social anthropology, and this clearly influences her analysis. Tett was one of the few journalists to warn of the impending crisis before it came to pass, and notes various events that are clear instances, or at least very close relatives of, normalization of deviance. Lewis's brilliant book tells the story of the few individuals who recognized that deviance, and bet on the markets failing. For more scholarly, academic, studies of the sociology of the financial markets, see the works of MacKenzie and his colleagues (MacKenzie 2008a, 2008b; MacKenzie *et al.*, 2008), although all of those pre-date the turmoil of the past three years. A strategic briefing paper written for the UK Government's Office of Science by Cliff (2010a) discussed the danger that normalization of deviance posed in high-frequency trading systems in the global financial markets, and the possibility of major catastrophe happening within very short time-scales; the final version of that paper was submitted to the government nine days before the Flash Crash.

One significant difference between previous financial crises and the Flash Crash is the speed at which they played out. In the past quarter of a century, financial-market trading has shifted from being a largely human, face-to-face activity, through a twenty-year stage of being phone-and-screen-based rather than face-to-face, but still largely requiring a human at each end of the phone or screen. Within the past decade, however, a fundamental technology-led shift has occurred. Increasingly, the counterparties at either end of the trade, at each end of the telecoms cable, are pieces of software rather than humans. Algorithmic trading systems are increasingly trusted to do trading jobs that were previously done by human traders, and to do jobs that would require super-human data-integration abilities in a person. Many automated trading systems have limiters and circuit-breakers built in to prevent major losses, but as was seen on May 6th, the system-wide interaction between multiple independently-engineered automated trading systems had at least one unknown catastrophic failure mode. A major proportion of traders in the markets are still human, but to understand today's markets it is necessary to study the interaction of these human traders with their automated counterparts.

The global financial markets, considered as a system, is made up of components, of constituent systems such as the various electronic exchanges and the automated and the manual trading systems operated by the various investment banks and hedge funds, all of which have been developed, procured, operated and managed independently. That is, the current global financial markets are, from a technology perspective, *systems of systems* (SoS). The effects of failure in one or more of the constituents may be contained, or may ripple out in a domino-effect chain reaction, analogous to the crowd-psychology of contagion. In this very definite sense, the global financial markets have become high-consequence socio-technical systems of systems, and with that comes the risk of problems occurring that are simply not anticipated until they occur, by which time it is typically too late, and in which minor crises can escalate to become major catastrophes at timescales too fast for humans to be able to deal with them. The extent to which the SEC/CFTC report attributes cause to a single rushed block-sale as a $4.1bn hedge as the trigger-event in the Flash Crash seems comparable to the way in which the *Challenger* accident investigation report identified failed SRB O-rings: there is a wider socio-technical perspective that should not be ignored, and which was already being pointed to by some authors prior to the events of May 6th 2010 (Haldane, 2009; Cliff, 2010a).

That the global financial markets have become large-scale complex IT-centric socio-technical systems is perhaps no surprise, given the wider context that IT systems have moved from back-office support (for payroll processing, say) firmly onto the critical path for very many enterprises and organizations, to the point where failure of the IT system can incapacitate an organization. For example, ten years ago a failure of the IT servers in a hospital would not have a major negative effect; whereas in the near future, once all data is digitized at the point of capture and integrated with patient's historical data before delivery in an appropriate form to a healthcare practitioner, then when a hospital's servers go down it will cease to be a functioning hospital and instead be a big building full of sick people, with highly trained professionals frantically tapping the touch screens on their PDAs/tablet-computers, wondering where the data went. Similar stories can be told, or are already plausibly foreseeable, in very many private-sector, public-sector, and defence organizations in most industrialized economies.

So, the concerns expressed in here about modern computer-based trading in the global financial markets are really just a detailed instance of a more general story: it seems likely, or at least plausible, that major advanced economies are becoming increasingly reliant on large-scale complex IT systems (LSCITS): the complexity of these LSCITS is increasing rapidly; their socio-economic criticality is also increasing rapidly; our ability to manage them, and to predict their failures before it is too late, may not be keeping up. That is, we may be becoming critically dependent on LSCITS that we simply do not understand and hence are simply not capable of managing. This is something that we summarize schematically in a single three-line graph, which we show in Figure 1.

The UK Research and Training Initiative in the Science and Engineering of Large-Scale Complex IT Systems (LSCITS) was started in 2007 as a national strategic investment with a primary aim being to foster the formation of a new community of researchers and practitioners with training and experience appropriate for dealing with future engineering dominated by LSCITS issues. In the following sections we give our definition of such systems, describe the structure of the UK LSCITS Initiative, summarize some of our past and current work, and talk about the Initiative's future plans.

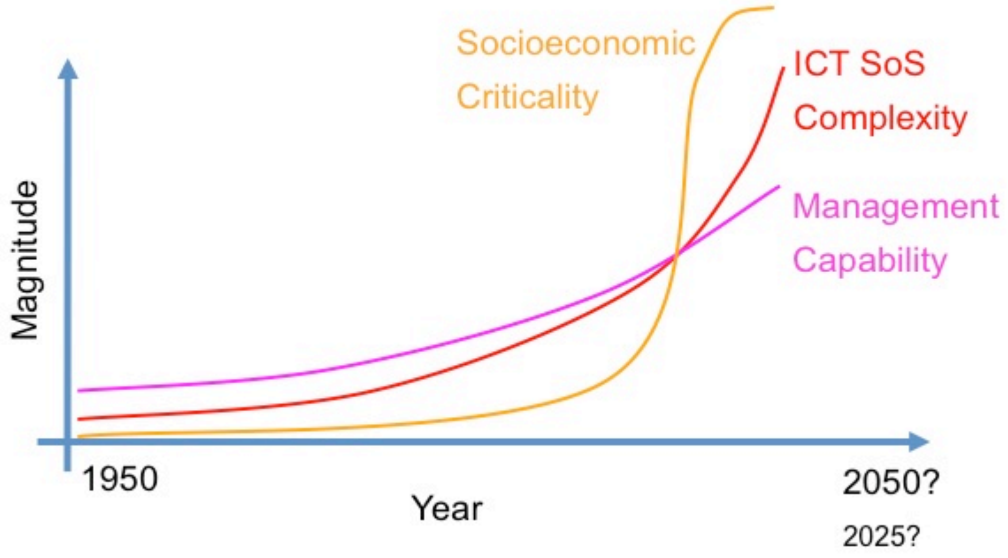

FIGURE 1: The LSCITS Complexity Crossover Crisis. The complexity of information and communications technology (ICT) systems of systems (SoS) has increased dramatically since ICT was first commercialized in the 1950s, and in recent years the socio-economic criticality of ICT SoS has also sharply increased, as very many enterprises and organizations in advanced economies have become dependent on the availability of ICT functionality as a key component on the critical paths of their operations. Over the same period, there is increasing concern (and growing evidence) that our ability to manage and predict the behavior of these critical ICT SoS is not increasing at the same pace, and so at some point in time there is the potential for crisis, where major socio-economic systems are critically dependent on ICT SoS whose complexity is beyond that which we can manage. We are deliberately non-committal on the precise timing of this crossover point: it could be a decade or more away, it could have happened already.

3. The LSCITS Initiative

3.1 Definitions

Before describing the structure of the LSCITS Initiative, it is worthwhile to first clarify some points of definition, as our interpretation of each of the words that contribute to the LSCITS acronym deserve some explanation.

Probably the most contentious letter in the acronym is the C for "Complex". What do we mean by complexity? Precise definitions of complexity vary quite widely across different domains and research fields, and a survey of the literature reveals tens of differing definitions (Bullock & Cliff, 2004, p.6). For us, we start with the observation that the word is often a short-hand term for the kind of mathematically nonlinear systems of interacting components commonly studied in "Complexity Science": systems where, at one level of analysis, the system can reasonably be described as a number of relatively simple entities, each interacting with some subset of other entities in relatively simple ways; but where there are nonlinearities either in the responses of the entities, or in the nature of their interactions, which compound across the entire system in such a way that the overall system-level behavior is difficult, or perhaps impossible, to accurately predict, even if given perfect knowledge of the system entities and their interactions; that is, the system-level dynamics can be described as exhibiting *emergent behaviour* and the system is a *complex system*. It's important to note that this definition of complexity rests in (temporal) dynamics rather than static, structural features. The structural complexity (or "complicatedness") of a system may be very high, confusingly more than a single human head can hold, but that does not necessarily qualify it as a complex system; and some complex systems are describable via only a small set of nonlinear differential or difference equations yet give rise to surprisingly rich emergent behaviour over sustained time-courses.[2] In fact, most LSCITS of significance are not merely complex systems, but are rather members of the interesting subset known as complex adaptive systems (CAS), where there are path-dependencies in the nonlinearities in the system which mean that the system's response to any given input or perturbation may change over time – colloquially we might say that the system adapts or learns or evolves over time, altering its behavior on the basis of past events.

Using a definition of Complexity and CAS that is rooted in the mathematics of nonlinear dynamics becomes less practicable as larger-scale socio-technical systems are considered. In many real-world large-scale systems, including systems composed from multiple constituent IT systems (i.e., SoS), the apparent complexity stems from a lack of understanding about the relationships between the constituents and their likely future dynamics. If the relationships between the constituents and/or their future behaviour can't be described or predicted economically and there are therefore gaps in our knowledge, then the system is in practice a complex one.

While it is important to be mindful of these subtleties in defining "complexity", it is also often convenient to have a short slogan that summarizes our meaning. We are happy to use this:

> *A system is complex when it is practically impossible to predict the consequences of a change to that system, with a high degree of certainty.*

[2] One of the simplest examples we can think of is Conway's famous "Game of Life" two-dimensional cellular automaton (Gardner, 1971), and many of the one-dimensional cellular automata discussed at length by Wolfram (2002).

Given this definition of complexity, our definition of "large-scale" can be expanded upon. While many LSCITS of practical interest are physically large and geographically disparate, these are not requirements of our definition, rather, for us we take large-scale to indicate that a large number of entities and interactions within the system may need to be taken into account in understanding or predicting the system-level behavior. For us, largeness-of-scale is a shorthand for systems whose scale is sufficiently large that they are subject to *normal failure*: where even if the component entities in the system are highly reliable, the sheer scale of the system means that the number of entities is so large that there will always be one or more that have failed. Say, for example, that a system is composed of entities that have "five nines" (99.999%) reliability, so any one entity would be expected to be unavailable for roughly five minutes per year; if the system is made up of 100,000 such entities, then at any point in time the probability of all entities being operational simultaneously is only 37%. That is, there is a 63% chance that one or more entities will have failed, and the system needs to be engineered to cope with this level of failures as "normal". A major consequence of normal failure in large-scale systems is that it results in systems that are significantly heterogeneous. As component entities fail, they are replaced by new units that do not necessarily exactly match the old failed components. So, for instance, even if a large-scale IT installation is equipped "from scratch" with 10,000 identical servers (a small number by current standards), within a relatively small period of time the occurrence of routine failures will mean that the population of servers is no longer entirely homogeneous, and the system needs to be engineered to accommodate or cope with this heterogeneity, and perhaps also to exploit positive aspects of it.

In fact, nontrivial homogeneity is to be expected in all LSCITS. For some, it will be because they started out as from-scratch homogeneous constructions and then accrued failures requiring patches and repairs, as just described. But in fact very few LSCITS are commissioned and built from scratch, starting with a clean sheet. For the vast majority of LSCITS the heterogeneity is a major factor *because* they were never designed from scratch: very many LSCITS grow organically, as existing systems are extended with new technologies, and as previously standalone systems, designed and developed and extended independently, are integrated to create new super-systems (i.e, systems-of-systems). In LSCITS, heterogeneity is a given.

To complete our account of what is an LSCITS, we note here that the presence of "ITS" (Information Technology Systems) as the last three letters in the LSCITS acronym is, in part, an accident of history – the LSCITS title on the Initiative pre-dates the direct involvement of any of us in it: LSCITS was the name chosen by the EPSRC, the UK public funding agency, when they decided to set up the Initiative, prior to any of us being directly involved. We treat "ITS" as a short-hand for high-consequence software-intensive socio-technical systems (but it has been pointed out to us that the acronym LSCHCSISTS is even less catchy than LSCITS, which seems a fair point).

For those already familiar with the excellent work of the Ultra-Large-Scale (ULS) Systems Group at the Carnegie-Mellon University Software Engineering Institute, it will be clear that there are very strong overlaps between our definition of LSCITS and the CMU SEI's definition of ULS Systems (Northrop *et al.*, 2006). We formed the LSCITS consortium and agreed our terms and aims entirely independently of the SEI ULSS team; only when we then formed the International Scientific Advisory Board for the LSCITS Initiative, one of our advisors alerted us to the similarity between our initiative and the ULSS team's recent report. There is strong, complementary alignment between the US ULS Systems work and the work of the UK LSCITS Initiative.

3.2 Structure

As will be clear from the text in the preceding section, our notion of LSCITS spans a spectrum from the mathematics of complex adaptive nonlinear systems, to the organizational processes that enable the engineering of complex socio-technical systems. The need for an interdisciplinary approach is manifest. We have structured the LSCITS Initiative to involve all the constituent research fields that we feel to be directly relevant; our primary interest is what happens at the intersections of these fields. The constituent fields are:

- *Complexity Science:* the study of nonlinear dynamical systems that exhibit emergent behavior, specifically complex adaptive systems (CAS), as introduced previously in our discussion of the definition of "complexity".
- *Predictable Software Systems (PSS):* the study of formal methods for computer systems engineering, involving the creation of models of existent or proposed systems, expressed in a formal language that can then be subject to automated theorem-proving and/or simulations for comprehensive search of the state-space. When the model has been proven or demonstrated to satisfy its requirements, it can serve as input to automated code-generators. PSS techniques show great potential, but their applicability in (ultra-)large-scale systems is limited by the poor scaling properties of current methods; improving their scalability is a current research issue.
- *High Integrity Systems Engineering (HISE):* recognizing that PSS approaches cannot currently fulfill all the needs of real-world applications where software is responsible for maintaining the integrity of safety-critical systems (such as aerospace systems, nuclear power stations, defence command & control), there is a distinguished body of engineering research and teaching that has been developed for engineering such high-integrity systems. The intellectual heritage of much of this work is traceable back to classical engineering control theory, with its separation of *plant* (the thing to be controlled) and *controller* (the thing that does the controlling); in consequence, historically HISE work addressed systems that have relatively simple control interfaces (buttons, dials, throttles) and considers the human users as essentially external to the plant-controller system. In many systems of significant interest, that approach is entirely appropriate. However, for many systems of systems, it is necessary to consider the human agents as components within the system, i.e. to adopt a socio-technical perspective. HISE work now embraces such issues, linking to work on social-technical systems and embracing the system of systems paradigm.
- *Socio-Technical Systems Engineering (STSE):* historically, the majority of research in socio-technical systems has been largely *a posteriori* descriptive, using techniques from human-computer-interaction, organizational and social psychology, and workplace ethnography, to formulate insightful analyses of existing systems. There is a less well-developed tradition of developing tools and techniques that are predictive and prescriptive; i.e., of developing *a priori* methods for engineering socio-technical systems. Extending socio-technical analysis techniques into software-systems engineering is a current research challenge.
- *Complexity in Organizations (CiO):* we believe it to be manifest that many LSCITS have come into existence to support large-scale networks of interacting groups or organizations. The interacting groups or organizations may be separate independent enterprises, or they may be divisions within some larger single enterprise. Here we take "enterprise" to include public sector organizations as

well as private firms; and the groups or organizations can be any size. Very often, the complexity in the LSCITS is present as a direct reflection of the complexity in the (network of) organizations, and understanding that organizational complexity is a deeply non-trivial problem. Thus, we have a commitment to work with organizational theorists, economists, and management and political scientists to understand these social aspects of socio-technical LSCITS.

- *Novel Computation Approaches (NCA):* finally, as the size of socio-technical LSCITS increases, many traditional engineering techniques do not scale well. Much engineering practice, that has served us so well for decades, is based on divide-and conquer: a process of recursive hierarchical decomposition is applied, breaking a system into subsystems, which are in turn decomposed into sub-sub systems, continuing until the system is decomposed to a sufficiently low level that the engineering of each component is tractable. Such hierarchical decompositions often result in engineered systems having hierarchical tree-structured control architectures, which scale badly because of the need for communication from the leaf-nodes up to the root node, this is a source of delays and noise, and the root node is a significant vulnerability, as a single point of failure. In the past two decades a variety of novel, decentralized, approaches to management and control of large-scale and distributed systems have been developed; many of them stemming from Complexity Science studies of CAS. We intend our Initiative's research to include explorations of the application of these novel decentralized approaches in LSCITS contexts. They are perhaps most relevant in the design, management, and control of ultra-large-scale IT facilities required for "cloud computing", discussed further in Sections 3.3.1 and 3.3.5.

We have acknowledged from the outset that there are unlikely to be productive lines of inquiry in trying to forge a single intersections between all of these constituent fields, nor even in trying to force intersections between every possible pairing of these subfields; nevertheless there is a coherent path through the fields and their intersections, linking from one end of the spectrum to another, as illustrated in Figure 2.

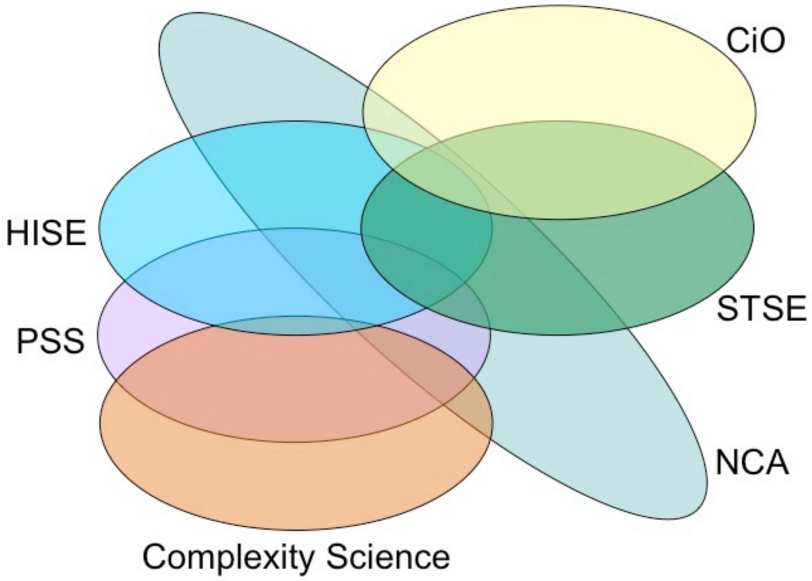

Figure 2: The "Stack" of overlapping research fields which constitute the LSCITS Initiative. The acronyms are defined and explained in the text.

There are other major UK publically-funded initiatives, each involving multi-millions of pounds of funding, that are relevant to LSCITS and that constitute significant investment in one or more of the fields shown in Figure 2. We illustrate the most relevant ones in Figure 3. There are three independent Doctoral Training Centres (DTCs) each training 40 or more PhD students over four or five years of student-intake: these are the Bristol Centre for Complexity Science (BCCS)[3], the Warwick Centre for Complexity Science (WCCS)[4] and the Southampton Institute for Complex Systems Simulation (SICSS)[5].

Issues of socio-technical systems engineering and understanding complexity in organizations are also being addressed by two UK "Engineering Doctorate" (EngD) centres focused on Systems Engineering. These are the Bristol/Bath Systems Engineering Doctorate Centre[6], and the Loughborough Systems Engineering Doctorate Centre[7]. The EngD is a degree fully equivalent to a traditional academic PhD, but much more focused to the needs of industry. The LSCITS Initiative's training programme includes over £4m for its own EngD programme, based at the University of York, the activities of which are discussed later in this document (see Section 3.3.6). Figure 3 illustrates the coverage of these other (non-LSCITS) DTCs and EngD Centres, relative to the "Stack" introduced in Figure 2.

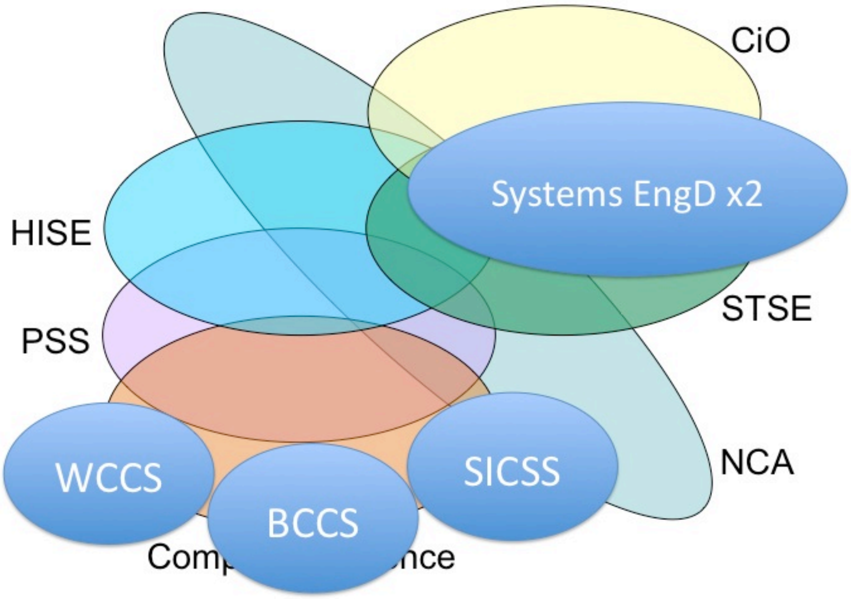

Figure 3: Other UK publically-funded doctoral training activities, with over £25m committed to training a total of more than 200 doctoral students, cover some aspects of the LSCITS Stack.

[3] http://bccs.bristol.ac.uk/
[4] http://www2.warwick.ac.uk/fac/cross_fac/comcom/
[5] http://www.icss.soton.ac.uk/
[6] http://www.bristol.ac.uk/eng-systems-centre/index.html
[7] http://www.lboro.ac.uk/departments/el/sedc/

Because of the very significant UK investment in Complexity Science doctoral research and training already underway, we took the view that the UK LSCITS Initiative should work with the WCCS, BCCS, and SICSS, rather than try to duplicate their efforts. For this reason, we are not actively investing our funds in complexity science research and training. Nevertheless, the LSCITS EngD training programme does require its students to pass a taught masters-level[8] module, developed specifically for the LSCITS EngD, which covers pertinent aspects of complexity science. Complexity science is not the only bespoke module that our LSCITS EngD students are required to complete: we have also designed similar masters-level modules for the other aspects of the LSCITS Stack, with the intent that graduates of our LSCITS EngD re conversant with *all* of the various disciplines and perspectives that we feel need to be combined in science and engineering of large-scale complex IT systems. The coverage of the LSCITS EngD is illustrated in Figure 4. We discuss the LSCITS EngD training programme further in Section 3.3.6.

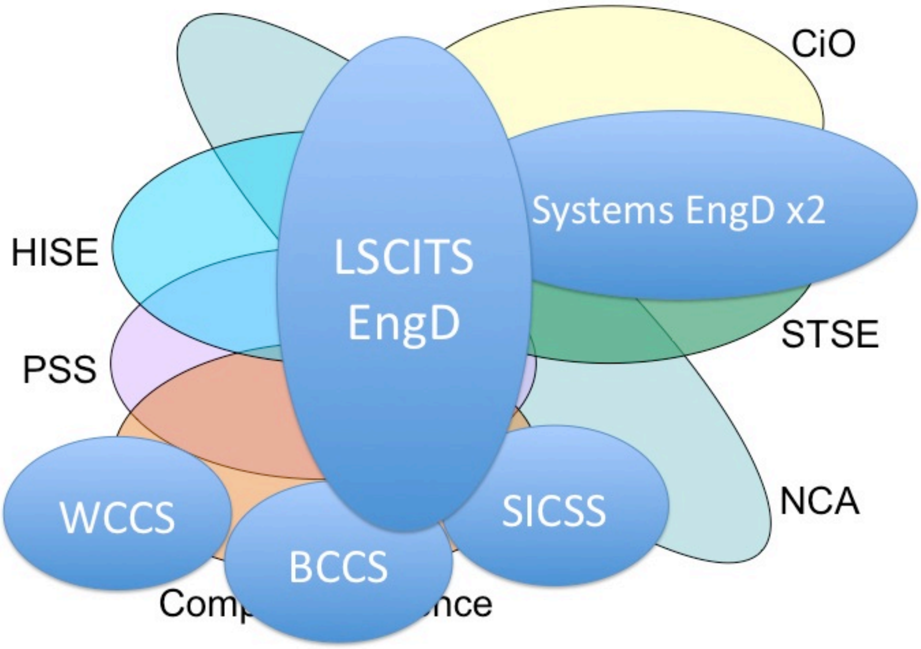

Figure 4: The LSCITS Engineering Doctorate (EngD) training programme requires its students to be exposed to all aspects of the LSCITS Stack, primarily by requiring them to complete taught masters-level modules specifically developed for the LSCITS EngD, covering these core topics.

[8] By "masters-level module" we mean a self-contained taught unit, typically requiring approximately 100 hours of student time, that contributes credits to a masters degree (e.g. MSc in the UK, or SM in the USA).

In our initial proposal to the UK Engineering and Physical Sciences Research Council (EPSRC), the primary funder of the LSCITS Initiative, we argued that the funds we sought would be used to establish a new community. Our specific words were:

> *The proposed Initiative's programme of work is intended to last for five years, but it is our intent that this be viewed as a period of pump-priming "ramp-up", establishing at steady-state a well-coordinated community of interacting researchers, self-sustaining by generating ongoing financial support from public funds and from industrial sponsorship and collaborations.* (Cliff *et al.*, 2006, p.2)

To this end, we have also engaged in a programme of dissemination, outreach, and "community building". The main LSCITS Initiative website[9] includes listings of our publications to date, and of the events we organize. We've given many invited seminars and conference keynote presentations describing the work of the LSCITS Initiative in a variety of contexts, including a series of lectures given to over 10,000 schoolchildren.[10] Our organized events have included a number of small workshops where academics from the Initiative and industry practitioners meet to discuss key issues; and a one-day public symposium showcasing the various strands of work underway within the Initiative. In addition to our website, we have set up an LSCITS group on the popular professional social-networking website LinkedIn,[11] which has approximately 300 members. To service this 300-strong community, we are launching a quarterly LSCITS Newsletter, written by an experienced technical journalist. The newsletter will be circulated as PDF and will carry news and updates from the LSCITS Initiative's researchers, interviews and case-studies from industrial partners, and summaries of wider technology-industry news stories. The first issue of the newsletter will be circulated to the LinkedIn LSCITS Group members in Q4 of 2010.

Having established the broad intellectual structure of the LSCITS Initiative, discussed its relationship to other major UK publically-funded academic initiatives, and described our community-building work, we now move on to explaining past and current research activities of the LSCITS Initiative, before concluding with discussion of our current strategic direction.

3.3 Research Programme

3.3.1 Autonomic Computing and Predictable Systems

We explained in the previous sections that IT systems are growing ever larger and more complex, and we described the costs and risks associated with this trend. Overcoming some of these costs requires IT systems capable of managing by themselves, with limited or no human intervention. Reducing the risks requires that they do so in predictable ways.

Ways to achieve these two requirements – self-management and predictability – have traditionally been explored in isolation, by different branches of Computer Science. The development of IT systems capable of self-management in the presence of changes in system state, workload and objectives has been the activity area of *autonomic computing*

[9] www.lscits.org

[10] see www.gcsesciencelive.net

[11] www.linkedin.com/groups?mostPopular=&gid=2484998

for the past decade. Ensuring that IT systems are predictable (in the sense that they satisfy certain classes of constraints and invariants) has long been the object of study for *formal methods*. A major effort within LSCITS has been directed towards integrating key results from the two branches of Computer Science for the first time, with particular focus on the use of quantitative automated verification of system features such as stochasticity, costs and time (Kwiatkowska 2007; Kwiatkowska *et al.* 2010). The outcome of this effort is described briefly in the remainder of the section.

Many LSCITS are systems with constitutents and components that are developed, procured, operated and managed independently – or *systems of systems* (SoS). We therefore started by identifying the software engineering techniques that have the potential to help address the challenges associated with SoS development (Calinescu & Kwiatkowska, 2010). The techniques identified by this study and explored in our subsequent work include formal analysis and verification; autonomic computing; machine learning; and model-driven development.

We have set out to address the issue of predictability by developing algorithms for systems that exhibit stochasticity and adversarial behaviour. Working with the model of probabilistic timed automata extended with price information, we formulated and analysed algorithmic problems that can deliver average, expected and optimum prices (Jurdzinski *et al.* 2009), also for the case of timed games (Forejt *et al* 2010). These methods have high complexity and therefore currently only applicable to small-scale systems. Future work is aimed at formulating efficient implementation techniques and scaling them to realistic systems.

The method for the development of predictable autonomic IT systems introduced by Calinescu and Kwiatkowska (2009a) uses run-time quantitative verification to perform dynamic reconfigurations guaranteed to achieve a range of performance- and reliability-related system objectives. The method employs Markov-chain quantitative analysis to dynamically adjust the parameters of an IT system in line with changes in its state, environment and objectives. Our toolset for the computer-aided development of self-star systems (Calinescu and Kwiatkowska, 2009b) uses this method and model-driven development techniques to significantly reduce the time and expertise required to build autonomic IT systems. The toolset has been used successfully by LSCITS Initiative team-members and by other external researchers, in application domains including quality-of-service optimization in service-based systems (Calinescu *et al*, 2010a), autonomic management solutions for the military domain (Werkman *et al,* 2010) and adaptive resource allocation for mobile robots (Hernando *et al.*, 2010).

Using existing formal techniques in online scenarios can augment complex IT systems with autonomic capabilities in predictable ways, but is feasible only for medium and small-sized systems (Calinescu *et al.,* 2010b). New techniques are needed to extend the applicability of the approach to large-scale complex systems, and the work that we carried out to identify them has produced several significant results.

We aim to improve the scalability of quantitative verification, opening up the opportunity for its use in large-scale autonomic systems, by means of compositional reasoning based on the *assume-guarantee* pattern. The technique works by decomposing the quantitative verification of large systems into separate analysis steps for each of its constituent components. This decomposition is made possible thanks to assumptions that the verification of one constituent makes about the properties of another. An independent analysis step is then carried out to confirm that the assumptions are justified, and thus to guarantee that the verification result is correct.

However, in the context of large-scale systems such assumptions are difficult to discover manually. In Feng *et al.* (2010), we introduced an automated technique for learning assumptions for compositional verification of probabilistic systems using machine learning. Future project work will explore this potential.

Using the right level of abstraction allowed Calinescu *et al.* (2010b) to verify a range of quality-of-service (QoS) requirements for service-based systems independently, by using different Markovian models. This allowed the implementation of autonomic quality-of-service (QoS) management and optimisation for larger systems than it would have been possible otherwise. As an extension of this work, we are aiming at developing a distributed QoS management technique for service-based systems of systems.

As online formal techniques alone are unable to address all needs of LSCITS, a separate strand of the Initiative's work has looked at hybrid approaches. In Kurd *et al.* (2009), we proposed a framework in which quantitative verification is employed alongside safety analysis techniques in dynamic risk management for aero-engine control. A hybrid approach to be investigated in the future involves supporting decisions in autonomic IT systems through a combination of formal verification and bounded-error heuristics. The formal verification component of the approach is envisaged to yield a close-to-optimal decision in a timely manner, and a faster technique can then be used to refine this decision.

Several of our PhD and EngD students are working on research that will contribute to increasing autonomy in the self-management of ultra-large-scale computing systems. Marco De Luca and Charlotte Szoztek are each exploring the development of new adaptive trading strategies that could be deployed in decentralized market-based resource-allocation and control of LSCITS (Cliff & Bruten, 1998; De Luca & Cliff 2011). Chris Musselle is exploring mechanisms, inspired by the human immune system, for increasing security in large-scale networked computer systems (Musselle, 2010), and Afnan Khan and Mustafa Aydin both recently commenced EngD research on issues in cloud-computing security, sponsored by British Telecom. Tom Cassey (part-sponsored by Hewlett-Packard) is working on statistically rigorous decision processes, motivated by their need in scheduling and resource-allocation in large data-centres. Duncan Tait (sponsored by Thales) and Mo Haghighi are each looking at aspects of large-scale dynamic wireless communication networks, and Ilango Sriram (sponsored by Hewlett-Packard Labs) is developing simulation models of cloud-scale data-centres that allow new management policies to be explored before deployment (Sriram, 2009; Sriram & Cliff, 2010).

3.3.2 High Integrity Systems Engineering (HISE)

The starting position of many of the traditional approaches to high-integrity systems engineering (e.g. safety engineering) is an attempt to circumscribe systems, their users, and the environment in which they are intended to operate. From this, activities such as hazard identification and risk management proceed – often with the objective and implicit belief that the system can be made safe (or secure) before being allowed to operate in the 'real world'. We contend that this approach is both hard to apply for LSCITS, and insufficient to provide genuine assurance of dependable operation.

Many LSCITS can be characterized as Systems of Systems (SoS) – i.e. systems made up of elements that themselves have sufficient independence of function and autonomy to be described as systems in their own right. SoS are rarely static in structure; instead they are constantly evolving in response to operational stimuli, social and technological change. In addition, human users are often forced to compensate for deficiencies in the interoperation and integration of SoS. These characteristics heighten the importance of identifying and planning for change as part of any systems assurance activity, analysing the (often informal) reliance on humans to mitigate shortfalls in the integrity of computer-based systems, and ensuring the practicality and scalability of regulatory oversight and acceptance activities. Our work on SoS safety, through collaboration with The UK National Health Service (NHS) Connecting for Health Team, has begun to explore how current clinical safety assessment is carried out on large-scale health informatics systems within the NHS. This work has so far involved a review of existing regulation and study of a current safety case for a UK Health Trust in-patient records system. Our aims in the ongoing work are two-fold. Firstly, we wish to draw upon our experience in the defence domain from the safety assessment of defence SoS – particularly, dynamic risk assessment utilizing Enterprise Architecture (e.g. MoDAF)[12] models, and the use of modular and incremental certification techniques to help handle issues of complexity and scale. Secondly, we wish to formalise how confidence is being established through the composition of evidence gained through design assurance, acceptance testing, service agreements, user trials, and in-service user feedback.

The complexity of many computer-based systems can also provide challenges in the prediction of in-service operational behaviour. One area we have studied where this is particularly the case is the domain of adaptive systems. Adaptation can offer the promise of resilience to a changing operational environment. However, it also suffers from problems of predictability and gaining *a priori* assurance of dependable operation. Past work in this area has often focused on the introduction of statically defined constraints to in-service adaptation (i.e. adaption 'within limits'). This can both limit useful adaption and suffers from the assumption of being able to predict future change drivers. In our work we have defined a framework that is based upon dynamically generated (risk-based) constraints where ultimately assurance relies upon the adequacy and integrity of the constraint generation, alongside the constraint-based limitation of in-operation adaptation (Kurd *et al.*, 2009).

One of the key themes in the above work is balancing flexibility and predictability. We are also investigating this theme in ongoing research on system engineering processes. There is a substantial tension when engineering complex systems in balancing flexibility and *control*: the engineering process needs sufficient controls in place to ensure that risks are suitably addressed. At the same time, flexibility in scheduling activities is beneficial for dealing with unexpected change requests and unanticipated requirements. Achieving a balance between flexibility and control is at the heart of so-called *agile methods* for software engineering. Though the term *agile* is overloaded and often misused (or misunderstood), at its core it refers to engineering processes that are iterative and incremental and that acknowledge that engineering and product requirements may be discovered as engineering proceeds. For software engineering, agile methods have been shown to help deliver high quality software, often faster than more traditional plan-driven approaches. One of the core research directions taken within the HISE research in the LSCITS Initiative is to attempt to achieve some of the attractive benefits of using agile methods when building high-integrity (particularly safety critical) complex systems. This is a challenging problem, as it involves overcoming technical difficulties (e.g., how to develop and deploy critical systems in an

[12] See www.modaf.com

incremental way) as well as organizational difficulties (e.g., how to convince certifying authorities of the acceptability of such processes[13]).

One concrete contribution that we have made in this area is by looking at how traditional HISE methods can be made 'more agile' through adoption of principles from agile methods; this is contrasted with attempting to make a wholesale adoption of an agile method. Our first step towards this was presented in (Ge, Paige, & McDermid 2010a), where we precisely defined the notion of *minimal up-front design* for a safety critical system: it is a design that is sufficient for applying hazard analysis techniques. At the same time, we showed how to deploy iterative development styles for critical systems, by precisely defining the concept of a *release* as one that includes an argument that the release is acceptably safe, and building on notions of modular safety cases. We trialed these concepts in an example, of building an altitude data display system, and ran a number of iterations that included construction of safety arguments. One of our conclusions is that, despite the inherent complexity in such a critical system, a traditional up-front design can be made *extremely simple* for the purposes of enabling a safety assessment.

A second concrete contribution that we have made in the same area is in the development of more flexible (agile) methods for safety assessment. Work that we carried out in 2009, on the development of *probabilistic failure propagation and transformation analysis* in collaboration with Oxford, was continued. We investigated the extent to which state-of-the-art probabilistic model checkers, like *PRISM*[14] (developed by members of the LSCITS PSS research group), could be used directly to support safety analysis (Ge, Paige, & McDermid 2010b). We developed transformations from complex systems architecture models to *PRISM* input modules, and demonstrated that *PRISM* checks could be used to encode specific kinds of safety analysis. Experiments were carried out to illustrate the applicability of the overall approach to different kinds of failure, e.g., processor crash rates on overall large-scale system behavior. The flexibility of *PRISM* and our transformations allowed us to easily run a series of *what-if* style analyses related to different system configurations, and different failure scenarios.

A third concrete contribution we have made has been exploring more *flexible (agile) approaches to development of an Enterprise Architecture* (EA). An EA provides mechanisms for describing business structures and processes connected to business structures. An EA can be helpful in finding effective ways to deploy and use complex ICT systems. However, producing an EA can be very expensive and time-consuming, and the resulting EA descriptions can be difficult to change. We have been working with hospitals in Beijing, China, to help understand the criteria that are of value to them in constructing *sufficient and adequate* models of EA, and sufficient and adequate documentation of those models. This work has only recently got underway, through a series of interviews (with hospital managers and IT managers) and hospital visits in Beijing. As a result, Enterprise Architectures are in the process of being developed and will be reviewed in concert with the stakeholders on the ground. Our objective is to develop guidance on improving the EA process itself, based on the principles of *agile documentation.* Future work will also explore better approaches to taking into account government regulations, and the tension between these regulations and commercial IT drivers. We have chosen to investigate Chinese hospitals with the intent that this will

[13] Experience shows that they are very conservative and, although standards and guidelines such as DO178B (EUROCAE, 1994) are objective-based, there tends to be a reluctance to accept anything other than the "standard" approach.

[14] www.prismmodelchecker.org

give us a comparator for work with the NHS in the UK, and perhaps some insights into the impacts of the different social and cultural environments in the UK and China.

Several of our PhD students, as well as our students on the LSCITS EngD, are working in the general area of high-integrity systems engineering, and thematically many of them are addressing this tension between flexibility and control. James Williams (PhD student on the LSCITS research programme) is looking at ways to bridge standardized approaches to high-integrity systems engineering with formal techniques (e.g., from the PSS theme), using model-driven engineering as a basis. James is particularly interested in scalability issues, and recently presented a paper at the European Conference on Modelling on a scalable back-end for supporting his approach to integration (Rose, Kolovos, Williams, *et al.* 2010). Jason Reich (PhD student on the LSCITS research programme) is researching formal verification of compilers and language processors (themselves complicated systems), but focusing on scalability and efficiency issues. Jason recently presented a paper at the Metacomputation conference in Russia on his new techniques for efficiency supercompilation (Reich, Naylor & Runciman 2010). Frank Burton (EngD student, sponsored by The Salamander Organization) is developing flexible techniques to support trade-off arguments in the context of through-life capability management. He intends to exploit model-driven engineering techniques, possibly through a set of coordinated and interoperable domain-specific languages, to enable the types of tradeoffs required by the defense industry (Paige, Burton, & Poulding 2010). Paul Mayo recently started an EngD (sponsored by SafeEng) exploring safety engineering techniques deployed by the UK Ministry of Defence; his work straddles the HISE and Socio-Technical Systems Engineering space. Similarly, Mick Warren has commenced an EngD (sponsored by AACE) focusing on understanding risk in software acquisition processes. Will Lunniss (EngD, sponsored by Rapita) very recently started his studies looking at on-target verification issues in LSCITS, with a particular emphasis on real-time issues. Alex Fargus (EngD, sponsored by Cybula) is investigating prognostics and diagnostics issues as they pertain to distributed Grid systems. In a similar vein, Hashem Gazzawi and Andrew Burkimsher, both EngD students sponsored by Airbus, are separately researching issues related to high-performance computing workflows in large-scale systems. Finally, Alex Healing (EngD, sponsored by British Telecom) is investigating very large-scale visualizations integrated with information management systems, with some particular interest in critical resource management.

As we indicated in Section 3.2, the work in HISE has a particular set of foci (safety, agile methods, SoS) but the scope of the doctoral (PhD and EngD) projects in the HISE domain is much wider.

3.3.3. Socio-Technical Systems Engineering (STSE)

The importance of socio-technical factors in the procurement, development, adoption and use of complex systems has been widely recognised and a range of socio-technical design methods have been developed such as ETHICS (Mumford, 1983) and Soft Systems Methodology (Checkland, 1981). However, while there have been successful projects using these methods, they have usually failed to be institutionalised after such projects and their use has often simply been forgotten. In the LSCITS project, we don't have the goal of developing new methods of socio-technical systems design - rather, we want to understand the barriers to the adoption of such approaches and to develop practical means to make it easier for companies to build on research in socio-technical systems and evolve their processes to take socio-technical considerations into account.

There are three types of barrier that we have encountered:

1. Lack of awareness. This is a real problem as systems engineering managers mostly have a technical background and have simply never been exposed to socio-technical considerations. To address this issue we have pulled together diverse strands of research in a major summary paper (Baxter & Sommerville, 2010) and are in the process of develop a Handbook of Socio-technical Systems Engineering (Sommerville *et al.,* 2010), which provides more detailed information and ready access to the literature on socio-technical systems.

2. Lack of tools and methods. Although there have been many different proposals for socio-technical design methods, many of them are difficult to use and lack tools for their support. Furthermore, they tend to be 'packaged' so that they can either be adopted as a whole or not at all. We are interested in how we can introduce socio-technical thinking into organizations and we recognize that this is most likely to be successful through evolution and through the use of familiar tools and methods. So, we have conducted a major study into social networking in organizations, with a view to understanding the attitudes of a range of stakeholders to this technology (Rooksby, 2010). We have carried out experiments in using social media for project support and we are carrying out investigations with industry (e.g. with users of ERP systems) that will allow us to propose how social analysis can contribute to their systems engineering processes.

3. Engineering culture. The engineering culture is, by and large, a 'can do' culture that aims to solve problems using ingenuity and technical knowledge. The culture is that problems have a technical solution and, in situations where systems are less successful than anticipated, the reasons are primarily technical or managerial – lack of control, lack of technical competence, lack of resources, etc. In fact, we believe that all complex systems are profoundly influenced by socio-technical considerations and that, in many cases, socio-political issues are far more significant than technical problems. Until we have a generation of decision makers who understand and accept this, change will be slow. In the LSCITS project, we are directly addressing this issue in the training programme: our PhD and EngD students are all being exposed to socio-technical issues and we hope this will influence their perspectives on how systems should be engineered.

Our work in socio-technical systems has always been rooted in an analysis of real problems in industry so a significant proportion of our effort is devoted to working with industrial partners to understand their problems. Such an understanding contributes to the work we are doing in transferring knowledge to industry and in developing new methods and techniques to address real industrial issues.

For example, one industrial partner, a major defence and aerospace supplier, suggested a research project to us that they felt would be of significant use to their government customers: simulating large populations of people interacting via technology, where a small number of those people are terrorists, plotting one or more attacks; the aim here is to use the simulation to better understand the "signature" signals generated by the terrorists' interactions that might be discoverable in the "noise" of the time-course of interactions in entire the population. There is a long-established literature on such simulations (e.g. Sloan, 1981), and our PhD student working on this project has started to make interesting discoveries.

Another example is our research interests in cloud computing, a significant proportion of which stem from our background in socio-technical systems. While we are convinced that cloud computing represents a major paradigm shift, we believe that the issues that affect the adoption of cloud technology are not simply technical. Rather, the migration of an organisation's computing capability to the cloud is influenced by human, social and organisational factors – in effect, it is a socio-technical problem.

Our work so far in this area has been concerned with case studies in industry to understand their concerns about using the cloud (Khajeh-Hosseini, Greenwood & Sommerville, 2010) and, from this, the development of a migration support system that takes socio-technical as well as cost considerations into account (Khajeh-Hosseini, Greenwood, Smith, & Sommerville, 2011).

We are about to start on a major development of our cloud work where we explore the problems of migrating application portfolios to the cloud. The problem here is that applications are both highly dependent on each other and are 'owned' by different parts of the organisation. Moving one application may have unanticipated effects and our aim is to explore modelling techniques that will allow us to include socio-economic as well as technical relationships between systems. We will use these techniques for portfolio management.

3.3.4 Complexity in Organizations: LSCITS in National Health & Social Care

LSCITS sit across a major divide, between the social and technological worlds. One of the challenges in the LSCITS Initiative is to explain why large scale IT systems and services diffuse rapidly in some organisational environments and slowly in others. Our work has led us down two avenues, one concerned with the meaning of two of our key terms, systems and complexity, and the other with the nature of organisations and of large scale IT networks.

The first avenue has involved reviewing the ways in which scientists – and in particular computer scientists – define the terms system and complexity. Viewed from a social science perspective, the use of the term "system" looks old-fashioned. Most social scientists would feel uncomfortable with the idea that organisations are information processing systems: there are many accounts that suggest that organisations survive and prosper *in spite of* communication failures. They would also take issue with the idea that organisation structures determine the behaviour of the people who work in them, as this implies that individuals are little more than automata. While organisations can be mind-numbing places at times, this assumption does not fit the available evidence: thankfully, individuals do matter.

Today, it is usual to acknowledge the role of information processing and of structures in organisations, but it is clear that these features cannot explain how organisations work. For the last 15 years there has been a major focus on knowledge creation: organisations can be viewed as knowledge-creating entities (Nonaka, 1994; Nonaka & von Krogh, 2009). There is also growing evidence that it is valuable to think about relationships, both within and between organisations, as being network-like in nature (Beinhocker, 2007). Relationships are rather more fluid than traditional ways of thinking suggest, with different networks of relationships being created and used for different activities at different times.

Some research in the complexity sciences offers us a way forward. The advantage of complexity science, for students of organisations, is that it does not make particular assumptions about information or about structures. Rather, the field starts 'bottom up', with agents interacting with one another and generating large scale behaviour (see, e.g., Epstein & Axtell, 1996). This has intuitive appeal, as organisations do seem to possess an 'organic' feel, evolving over time, not according to a blueprint or in pursuit of particular objectives, but growing ever more complex in response to external influences such as new technologies arriving or new markets opening up.

Viewed from the social sciences, though, many of the results generated within the complexity sciences deal in generalities, in abstractions, which sieve out most of the phenomena that interest social scientists. More positively, complexity science also points to possibilities that are not currently being discussed in sociology or political science. The idea that organisations – or their constituent elements – are self-organising is of particular interest, as social scientists do not have clear ideas about the self-sustaining nature of organisational life.

Putting the arguments about systems and complexity together, we propose that it is useful to view organisations as hybrids, as admixtures of 'traditional' control systems and complex systems. It would be very odd to reject the idea that organisations allow us to control a wide range of human activities. But equally, perfect control will always elude us: this is not because organisations are 'noisy' systems, but because they are complex, in the sense that they are in part self-organising, and their behaviour and future structures are unpredictable. This view is broadly consistent with work in the HISE stream, reported above, from which we can also draw the observation that large scale IT networks are also hybrids, balancing control and agility. (This usefully sheds light on an observation in sociology, which is that IT networks have dual properties, seemingly promoting control and increasing complexity at the same time.)

The second avenue of research has focused on a specific domain, health and social care. In order to overcome the problems associated with the abstract nature of the results in much complexity science it is important to ground research in specific settings. This allows us to combine insights about complex behaviours with other, more 'grounded', evidence. For example in our principal application area, health and social care, we can observe that progress with implementing large scale IT networks has been slow, compared to other sectors of the economy. We can also observe that there are consistent patterns in those IT networks which have been successfully implemented, which suggest homophily – successful systems are, in the main, ones which link together professionals with similar backgrounds, or where there is a critical professional dependency. For example, pathologists send test results to hospital consultants and to community physicians, and increasingly GPs can order prescriptions from pharmacies. But there is marked reluctance, in many countries, to share information about patients across professional boundaries. Amazingly, at least to anyone outside health care, there are hospitals where doctors have access to expensive and sophisticated electronic records systems – and nurses in the same hospitals use paper records.

We have pursued three topics in the area of health and social care, which will be discussed in a report on future directions for the development of large scale IT networks (Keen *et al.*, forthcoming 2011). The first is innovation. In health and social care, where progress with large scale IT systems has been limited in most countries to date, a central puzzle is: why so slow, compared to other sectors? Some of our work has looked for frameworks that can help to shed light on the problem. The second area is privacy, which we have investigated because it bedevils the use of large-scale networks in practice. We have drawn on socio-legal and bioethical literatures, both of which make the point that privacy is an 'asserted right': it does not have a secure legal status, seemingly in any country. Partly as a result, it offers a relatively weak basis for the development of robust rules for the handling of personal data. Our work suggests that a different approach, based on the concept of confidentiality, may prove be more practical. This approach would focus on ensuring that, while our names, addresses and other data are already publicly available, it remains important to protect any information generated in a consultation with a doctor or nurse.

The third area is policy making. If large scale IT systems are complex systems, or hybrids of complexity and control, then what are the implications for health and social care policies? Our work suggests that policy makers in many countries are still wedded to 'control concepts', believing that they can use IT networks to reduce costs and improve the safety and quality of services. If those networks are – even in part – complex systems then they will generate unpredictable behaviour. In other words, it does not make sense to assume that cost or quality improvements will follow successful implementation. There is good suggestive evidence that this is the case, but policy makers continue to under-estimate the importance of complexity. The challenge ahead is to devise policies that achieve the advantages of better control that IT can bring, while minimising the inevitable problems that they will throw up, simply because they are also complex, and some of their effects unpredictable.

In July 2010, the UK's new coalition government published a whitepaper with the title Equity and Excellence: Liberating the NHS,[15] that laid out plans for a major restructuring of the UK National Health Service. One major consequence of these plans is that the NHS Information Centre (NHS-IC)[16] will be given statutory powers to become the single national central repository for health and social care (H&SC) data on all NHS patients in England and Wales, with responsibility for collecting, curating, integrating, analyzing and summarizing data concerning "consumers" (patients/clients) and "producers" (doctors, hospitals, etc) of H&SC services. The NHS-IC's own description[17] of their role reads as follows:

> *We collect, analyse and present national data and statistical information in health and social care. … We have a pivotal role to play in delivering:*
>
> - *Trusted information critical to decision making and public accountability.*
> - *Standardised measures and national comparisons which are key to improvement.*
> - *Savings through the rationalisation of information services from other parts of the system.*

This presents a number of major socio-technical challenges, and manifestly makes the NHS-IC one of the most significant LSCITS in the UK, if not the world. Members of the LSCITS Initiative from each of the six partner universities are collectively in the early stages of working with NHS-IC staff on a number of research collaborations expected to be of value to the NHS-IC directly, and potentially also to the UK Government's Department of Health. These include: studies of the roles that cloud computing may play in the NHS-IC national data repository, and developing a metadata infrastructure for the repository; exploring automated market-based resource allocation mechanisms in the NHS; investigating issues of NHS data generation, validation and repair, quality and safety; studying the relationship between NHS measurement and indicators, and prior work on evidence and argumentation metamodels; studying issues of data management and organizational change; and quantifying the cost of change. All of these research interactions are currently in very early stages, and we expect to be able to report on them in forthcoming publications.

[15] www.dh.gov.uk/en/Publicationsandstatistics/Publications/PublicationsPolicyAndGuidance/DH_117353
[16] www.ic.nhs.uk
[17] www.ic.nhs.uk/about-us/more-about-us

3.3.5 New Work: Cloud Computing, Financial Markets

At the time of writing this paper, two new strands of work are commencing within the Initiative: one is a significant expansion of our efforts on cloud computing; the other brings us back to the events of May 6th 2010, the Flash Crash.

"Cloud computing", where computing resources are remotely accessed in a scalable, elastic fashion, with utility-style billing (see e.g., Cliff 2010b for an introduction), was a phrase not in common usage at the time that we wrote our initial proposal for the LSCITS Initiative, in the summer of 2006. Nevertheless, while the phrase "cloud computing" does not appear in that proposal, there was explicit discussion of the need for research on the management and control of such utility-style large-scale IT systems (Cliff, *et al.*, 2006, p.19); and the significance of utility computing as the likely next major transition in the commercial provisioning of IT had been discussed in a 2004 UK Government strategic briefing report (Bullock & Cliff 2004, p.3) that played a role in establishing the argument for the Initiative to be funded.[18] In October 2010 the LSCITS teams at Aston, Bristol, and St Andrews commenced on a new three-year collaborative project within the Initiative, hiring three new post-doctoral research fellows and two new PhD students, to conduct research on future cloud computing systems as large-scale complex IT systems, exploring issues at the infrastructural and socio-technical levels (Calinescu, Cliff, & Sommerville, 2009). This project builds on prior work in cloud computing that we had been pursuing in the first three years of the Initiative: much of the research on autonomic and predictable systems in Section 3.3.1 can be considered as enabling for ultra-large-scale data-centre installations; our prior research on socio-technical aspects of cloud computing was discussed in Section 3.3.3; LSCITS PhD students at Bristol and St Andrews collaborated on producing two review papers (Khajeh-Hosseini, Sommerville, & Sriram, 2010; Sriram & Khajeh-Hosseini, 2010); major companies active in cloud computing sponsor our doctoral students: we have one IBM-sponsored EngD student, two EngD students sponsored by British Telecom, and three PhD students sponsored by Hewlett-Packard, who have started to produce peer-reviewed publications (e.g. Sriram, 2009; Sriram & Cliff 2010, Rogers & Cliff 2010, 2011); and we were commissioned by Becta, the UK agency responsible for advising on IT in schools, to write a non-technical paper exploring the possibilities for cloud computing usage in UK schools education (Cliff, 2010b).

Finally, returning to the opening discussion of the Flash Crash, the briefing paper (Cliff, 2010a) that was submitted to the UK Government's Office of Science and Technology (OST) in late April 2010 played a contributory role in the OST deciding to set up a two-year project run by their *Foresight* unit[19], commencing October 2010, to investigate possible future scenarios in the technology-enabled global financial markets, exploring issues of systemic risk, viewing the markets as large-scale complex socio-technical systems. One of us (Cliff) is a member of the "Lead Expert Group" for this project, composed of a small number of senior regulators, industry experts, and academics.[20] We expect that others of us will be likely to contribute our expertise in high-integrity systems, socio-technical systems, and predictable software systems, to this project as it progresses. One long-term research ambition that we collectively hope to contribute to is the development of methods for predicting failures in LSCITS with sufficient forewarning that appropriate remedial actions can be taken without major loss of

[18] Another report that was also influential in establishing the case for the LSCITS Initiative was published by the Royal Academy of Engineering in April 2004 (RAEng&BCS, 2004).

[19] www.bis.gov.uk/foresight

[20] www.bis.gov.uk/foresight/our-work/projects/current-projects/computer-trading/lead-expert-group

system function (i.e., without shutting large chunks of the system down). If such methods can be developed, a related aim would then be to give quantitative *a priori* assurances concerning the likelihood of such events. Quantitative risk assessment techniques developed for safety-critical application domains such as nuclear power and aerospace engineering (see e.g. Stamatelatos *et al,* 2002a*,* 2002b; Hubbard, 2009; Dezfuli, *et al.,* 2009) are promising, but their extension to large-scale complex socio-technical software-intensive systems remains an open challenge.

3.3.6 The LSCITS Engineering Doctorate (EngD) Training Programme

The primary activity of the LSCITS Training initiative is the LSCITS Engineering Doctorate (EngD) Centre, based at the University of York. An EngD is an industry-based four-year doctoral programme. Students on the programme – known as *research engineers (REs)* – have industry sponsors from the LSCITS space, who direct and guide the engineer's research studies to deliver results that have both industrial value and doctoral-level academic merit. The REs are based at their industry sponsor for a substantial period of time, to help ensure the REs have opportunities to understand and experience the organizational structure and industrial mindset; at the same time, the research that the RE produces can be more easily transferred to the organization *in situ.*

Over the course of the four-year programme, research engineers complete a number of taught modules (including core LSCITS teaching on socio-technical systems, high-integrity systems, technology innovation, practical software specification, systems engineering and empirical methods) and, as indicated above, there are further taught modules which enable the REs to receive education on topics outside the scope of the LSCITS research programme. The REs carry out a substantial research project on topics that are of interest and value to their sponsor.

The LSCITS Initiative's main research programme (as discussed in Sections 3.3.1 To 3.3.5), although broad, is necessarily focused on a relatively small number of key topics where there remain significant research challenges. The LSCITS EngD RE projects and theses are also a major constituent of our overall research activities, but it differ in two significant respects. First, the EngD programme allows us to address issues outwith the scope of the main research programme which are pertinent to the development and assurance of real-world LSCITS, e.g. security engineering and accreditation, to provide an adequate grounding for the REs. Second, it can address issues within the scope of the LSCITS Stack where there are established solutions, which are therefore not explored in the main research programme, e.g. architectural frameworks such as MoDAF or Zachman (e.g. Sowa & Zachman, 1992). Thus in these respects the *taught* part of the EngD is wider in scope than the LSCITS research programme; this is currently also true of the research work undertaken by the LSCITS REs, and we expect this to remain the case.

The York LSCITS EngD Centre has recruited, from mid-2009, a large number of research engineers to work with companies and on projects that span the entire LSCITS space, e.g., from hard verification problems, to socio-technical problems, to organizational problems; many of these EngD projects have been mentioned earlier. Recruitment of further research engineers across the LSCITS space is actively continuing.

3.4 Strategy

3.4.1 Introduction

Here we articulate our current high-level strategy, which serves both to guide our future work and to give a more concrete basis for interacting with industrial partners.

The approach that we're currently pursuing is to take a System of Systems (SoS) view of LSCITS, and to focus on the need to achieve dependability and support rapid change. Whilst this doesn't cover all systems which can be characterized as LSCITS, it is a substantial enough and challenging enough subset of the "LSCITS problem" to be a useful driver for the remainder of the overall LSCITS Initiative

The notes in this section briefly expand on the problem-statement and then identify a number of strands of research that we believe could and should be developed in LSCITS, together with some links to other ongoing research programmes.

3.4.2 Problem Space

Here we identify three key aspects of LSCITS that we believe characterize an interesting and significant sub-problem, one of which requires a radical solution approach:

- *SoS:* many LSCITS are best considered as SoS. Separate systems are (in principle) designed by different authorities, and operated by different authorities, but they are combined with organisations and users to form the LSCITS, so the LSCITS is a SoS. The constituent systems will have some shared but also some conflicting objectives, and will enter and leave the SoS at different times. Some SoS may be intentionally designed, but it is likely that many SoS-LSCITS will simply evolve or emerge as organisations work together;
- *Dynamic:* the constituent systems in the SoS evolve at different rates, the usage of the systems, in terms of organisations and individuals, changes at different times, and the broader environment changes within times short with respect to the timescales necessary to meet a given objective or to make engineering design changes to a system;
- *Dependability:* users or organisations depend on the SoS for safety, security, availability, etc., and there will usually be trade-offs between these different properties to enable the SoS to satisfy its stakeholders.

We treat LSCITS as socio-technical SoS, embedded in organizational, cultural, regulatory, and political settings and our focus is on understanding this context to help design the SoS and its organizational impact, not on the (re-)design of organizations per se.

3.4.3 Speculative Solution Strategy

Our current solution strategy has two elements:

- How individual systems in the SoS are constructed, and
- How the SoS operates to satisfy stakeholders, meet dependability objectives, etc.; unusually this includes how dependability is assured.

We start off by considering the operational aspects, as this gives a context for analyzing how systems are constructed. Some assumptions are made about the issues of interfacing the systems in the SoS; these are set out here using the SPIT (Social-Process-Information-Technology) model, considered bottom-up:

- *Technology*: concerned with physical communications media, protocols, etc. – whilst there may be issues here it is assumed that these are outside the scope of the LSCITS programme, and these can be addressed by IPV6, DNS, etc., or other research initiatives.

- *Information:* concerned with the data passed between systems and its interpretation. Whilst schemas and meta-models, etc. can be used where the systems are pre-defined to work together, it is unclear whether or not this is sufficient to enable the dynamic behaviour outlined above; aspects of the dynamic interpretation of data fall within the scope of LSCITS.
- *Process (business):* concerned with the processes by which value is obtained from the SoS. Here issues of business process interaction and their dependability are key; this is assumed to be a core aspect of the LSCITS work, and must deal with the dynamic (unplanned) interaction of systems.
- *Social (organisational):* concerned with the context in which the SoS operates, including the organisations and stakeholders who determine whether or not the SoS delivers value; within the scope of LSCITS from the point of view of how value is judged, and how trade-offs are made.

We hypothesise that, in order to meet the problem-space challenges, the constituent systems in the SoS need to be "self-aware" and, perhaps, autonomic as they should be able to "negotiate" aspects of their behaviour to optimise the value delivered to stakeholders. In talking of self-awareness, we mean that the systems should carry their own dependability cases (in a manner analogous to proof-carrying code) and also be able to reason about those cases. This view is obviously an idealization: it would require a significant advance on the state of the art to deliver this in future green-field[21] SoS, and the issue of how such self-awareness could be retro-fitted into legacy, brown-field, SoS, is much more problematic. Either way, the dependability cases would be socially constructed and agreed documents, and may need to refer to (or make assumptions about) the broader socio-technical environment, so delivering this hypothesized solution strategy is clearly a socio-technical problem too.

Nevertheless, there is work to build on, e.g. in the UK Software Systems Engineering Institute (SSEI)[22] and the joint US-UK International Technology Alliance in Network and Information Science (ITA)[23]. To fully explore the view taken here, it would be highly desirable to work with a case study, and to start with simple methods and to build up more complex approaches over time.

In terms of construction there is need both to understand how to develop new software, and to deal with brown-field situations. There is already an established body of work on exploring agile development and dependability. What is proposed here is an assessment and refinement of this work, and adoption of ideas how to scale agile and lean development, to also consider how to develop dependability cases in an agile or lean manner. This might, for example, build on work on evidence-based development (EbD: Dick, 2009) or Graydon & Knight's Assurance Based Development (ABD) approach (Graydon, Knight, & Strunk, 2007). This will need to include development of the "operational" dependability cases needed by the SoS, and perhaps the development of evidence-based, or argument-driven, design. Scalability is certainly a key issue here, and

[21] In their 2008 book *Eating the IT Elephant: Moving from Greenfield Development to Brownfield*, senior IBM staff Richard Hopkins and Kevin Jenkins made the analogy between the greenfield/brownfield distinction in civil engineering, and modern-day large-scale complex IT projects. A greenfield engineering project is one in which construction takes place on a previously undeveloped site, allowing a "clean-sheet" approach at the design stage, with relatively little preparatory work required on-site before construction, and with relatively few constraints on the construction process. A brownfield project is one in which the site has previously been built on and hence may require significant clearing operation before construction, with the possibility of the added complexity from the requirement that existing structures must be retained and their viability maintained during the construction phase.

[22] www.ssei.org.uk

[23] www.usukita.org

there may be value in studying trade-offs between the complexity of individual systems and the complexity of the integration problems.

3.4.4 Case Study

A unifying case study would be valuable. One place to look for a case study is in the military domain where SoS are being constructed and there is considerable reuse of legacy systems, i.e. brown-field development, but it is unclear that it will be possible to access a good example. However, as a new element of the LSCITS work, based on cloud computing, has recently been initiated there would be considerable merit in finding a cloud-based example to use as a case study. Our nascent interaction with the UK NHS Information Centre offers the possibility of exploring the application of our strategy in the context of national-level health & social care record-keeping and data-analysis.

3.4.5 Observations

What is set out here is deliberately high level, but we felt it better to keep our "vision statement" brief. Perhaps the vision can best be characterized as:

- Systems in a SoS should be self-aware, in terms of managing their own dependability (and helping to establish a 'live' dependability case), and negotiating dependability as the SoS configuration changes;
- Individual systems are constructed rapidly using agile/lean techniques if new, and "system identification" techniques if brown-field, and the approaches produce the dependability cases to support SoS operation.

This cannot be adequately addressed with the current resources available within the LSCITS Initiative; but with some refocusing and with the support of industrial partners (e.g. through sponsoring students on our EngD and/or setting up industrial PhD projects), we would have the opportunity to refocus the work and thus to produce a more coherent programme around this vision. This can be further complemented by drawing on other programmes which LSCITS partners are involved in, such as the SSEI and ITA.

4. Summary

The problems posed by attempting to manage and engineer large-scale socio-technical systems are becoming ever more clear, but further research is needed to develop appropriate tools and techniques. Issues of scale, normal failure, organic growth, and emergent behavior have to be addressed. The UK LSCITS Initiative, like the US ULS Systems Initiative, is a step toward developing a new community of practitioners and researchers who are conversant with all the necessary subfields that can contribute to addressing issues in the science and engineering of such systems. The engineering of large-scale complex IT system is in its infancy, has some significant differences from engineering smaller-scale systems, and developing rigorous trusted approaches may be a long haul; we welcome the involvement of any researchers, practitioners, or sponsors who would like to become involved.

Acknowledgements

The LSCITS Initiative is funded primarily by the UK Engineering and Physical Sciences Research Council (EPSRC)[24] under grants EP/F001096/1, EP/F501374/1, and EP/H042644/1. Dave Cliff's early work on scoping the LSCITS Initiative and assembling the consortium was funded by EPSRC grant EP/E009263/1. The total funding for these four grants is £10,227,329, for which we are very grateful. We also thank the various sponsors of our PhD and EngD students: AACE, Airbus, British Telecom, The Bath/Bristol Systems Engineering Doctorate Centre, The Bristol Centre for Complexity Science, Cybula Ltd, Hewlett-Packard Labs, IBM, Rapita Systems, SafeEng, The Salamander Organization, The Scottish Informatics and Computer Science Alliance, and Thales. We thank John Rooksby and Linda Northrop for their comments on earlier versions of this paper.

[24] www.epsrc.ac.uk

References

G. Baxter & I. Sommerville (2010). 'Socio-technical systems: From design methods to systems engineering'. Interacting with Computers, doi:10.1016/j.intcom.2010.07.003

E. Beinhocker (2007) *The Origin of Wealth: Evolution, Complexity, and the Radical Remaking of Economics.* Harvard Business School Press.

S. Bullock & D. Cliff (2004). Complexity and Emergent Behaviour in ICT Systems. Foresight strategic briefing paper, for UK Government Department of Trade and Industry. Hewlett-Packard Labs Technical Report HPL-2004-187. Available from: http://www.hpl.hp.com/techreports/2004/HPL-2004-187.pdf

R. Calinescu, & M. Kwiatkowska (2010). Software Engineering Techniques for the Development of Systems of Systems. In: Choppy, S. and Sokolsky, O. (editors), *Foundations of Computer Software: Future Trends and Techniques for Development*, vol. 6028 of LNCS, pp. 59-82, Springer.

R. Calinescu, S. Kikuchi, & M. Kwiatkowska (2010a). Formal Methods for the Development and Verification of Autonomic IT Systems. To appear in: Cong-Vinh, P. (editor), *Formal and Practical Aspects of Autonomic Computing and Networking: Specification, Development and Verification*, IGI Global.

R. Calinescu, D. Cliff, & I. Sommerville (2009). *Cloud Computing for Large-Scale IT Systems.* Research proposal to the UK Engineering and Physical Sciences Research Council; submitted September 2009, commenced May 2010. Further details available at: http://gow.epsrc.ac.uk/ViewGrant.aspx?GrantRef=EP/H042644/1

R. Calinescu & M. Kwiatkowska (2009a). Using Quantitative Analysis to Implement Autonomic IT Systems. In: *Proc. 31st Intl. Conf. Software Engineering (ICSE'09),* pp. 100-110.

R. Calinescu & M Kwiatkowska (2009b). CADS*: Computer-Aided Development of Self-* Systems. In: Chechik, M. and Wirsing, M. (editors), *Fundamental Approaches to Software Engineering (FASE 2009)*, vol. 5503 of LNCS, Springer, pp. 421-424.

R. Calinescu, L. Grunske, M. Kwiatkowska, R. Mirandola, & G. Tamburrelli (2010b). Dynamic QoS Management and Optimisation in Service-Based Systems. To appear in: *IEEE Transactions on Software Engineering*.

CFTC & SEC (2010a). *Preliminary Findings Regarding the Market Events of May 6th, 2010.* Report of the staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory issues. May 18th 2010: http://www.sec.gov/sec-cftc-prelimreport.pdf;

CFTC & SEC (2010b). *Findings Regarding the Market Events of May 6th, 2010.* Report of the staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory issues. September 30th, 2010. http://www.sec.gov/news/studies/2010/marketevents-report.pdf

P. Checkland (1981). *Systems Thinking, Systems Practice.* Chichester, UK: Wiley.

D. Cliff & J. Bruten (1998). Simple Bargaining Agents for Decentralized Market-Based Control, in: *Proceedings of the 12th European Simulation Multiconference on Simulation -*

Past, Present and Future. Pp.478–485. http://www.hpl.hp.com/techreports/98/HPL-98-17.pdf

D. Cliff, J. Keen, M. Kwiatkowska, J. McDermid, & I. Sommerville (2006). *Large Scale Complex IT Systems Research Programme.* Research proposal to the UK Engineering and Physical Sciences Research Council; submitted December 2006, commenced April 2007. Further details available at:
http://gow.epsrc.ac.uk/ViewGrant.aspx?GrantRef=EP/F001096/1

D. Cliff (2010a). Networked Governance in the Financial Markets. Foresight strategic briefing paper, for UK Government Office of Science & Technology, Department of Business, Innovation, and Skills. Available from:
http://www.cs.bris.ac.uk/home/dc/Foresight_NetGov_v2a.pdf

D. Cliff (2010b). Remotely-Managed Services and 'Cloud Computing'. Emerging Technology for Learning Report, Becta (The British Education and Communications Technology Agency). Available from:
http://emergingtechnologies.becta.org.uk/index.php?section=etr&rid=15278

M. De Luca & D. Cliff (2011). Agent-Human Interactions in the Continuous Double Auction, Redux. To appear in *Proceedings of the 3rd International Conference on Agents & Artificial Intelligence (ICAART 2011).* January 2011, Rome.

H. Dezfuli, K. Dana, *et al.* (2009). *Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis*. NASA SP-2009-569:
http://www.hq.nasa.gov/office/codeq/doctree/SP2009569.pdf

J. Dick (2009). *Evidence-based Development: How to reduce risk, improve quality, and ensure compliance.* White paper available from Integrate Systems Engineering; pdf here:
http://www.integrate.biz/downloads/EbD_White_Paper_May_09.pdf

J. Epstein & R. Axtell (1996) *Growing Artificial Societies: Social Science from the Bottom Up.* MIT Press.

EUROCAE (1994). ED-12B/DO-178B: Software considerations in airborne systems and equipment certification. EUROCAE.

L. Feng, M. Kwiatkowska & D. Parker (2010). Compositional Verification of Probabilistic Systems using Learning. In: *Proc. 7th Intl. Conf. Quantitative Evaluation of Systems (QEST 2010)*, pp. 133-142.

M. Gardner (1971). On Cellular Automata, Self-Reproduction, the Garden of Eden, and the Game 'Life', *Scientific American,* 224(2):112-118.

X. Ge, R.F. Paige & J.A. McDermid (2010). An Iterative Approach for Development of Safety-Critical Software and Safety Arguments. In: *Proc. Agile 2010*, IEEE Press, August 2010.

X. Ge, R.F. Paige & J.A. McDermid (2010). Analyzing System Failure Behaviours with PRISM. In: *Proc. Fourth IEEE International Conference on Secure Software Integration and Reliability Improvement*, IEEE Press, July 2010.

P. Graydon, J. Knight, & E. Strunk (2007). Assurance Based Development of Critical Systems. In *Proceedings of the 37th Annual International Conference on Dependable Systems and Networks*, Edinburgh, U.K.

A. Haldane (2009). *Rethinking the Financial Network*. Text of a speech given at the Financial Student Association, Amsterdam, April 2009. Available from here: http://www.bankofengland.co.uk/publications/speeches/2009/speech386.pdf

A. Hernando, R. Sanz, & R. Calinescu (2010). A Model-Based Approach to the Autonomic Management of Mobile Robot Resources. In: *Proc. 2nd Intl. Conf. Adaptive and Self-adaptive Systems and Applications* (ADAPTIVE 2010).

R. Hopkins & K. Jenkins (2008). *Eating the IT Elephant: Moving from Greenfield Development to Brownfield.* IBM Press.

D. Hubbard (2009). *The Failure of Risk Management. Why it's Broken and How to Fix It.* John Wiley.

J. Keen *et al.* (forthcoming 2011) report on LSCITS in national-scale health & social care.

A. Khajeh-Hosseini, D. Greenwood, J. Smith & I. Sommerville (2011). The Cloud Adoption Toolkit: Supporting Cloud Adoption Decisions in the Enterprise. To appear in: *Software, Practice and Experience.*

A. Khajeh-Hosseini, D. Greenwood & I. Sommerville (2010). Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS. *Proc. 3rd IEEE Conf. on Cloud Computing.* http://doi.ieeecomputersociety.org/10.1109/CLOUD.2010.37

A. Khajeh-Hosseini, I. Sommerville & I. Sriram (2010). *Research Challenges for Enterprise Cloud Computing.* Unpublished technical report, http://arxiv.org/abs/1001.3257

C. Kindleberger (2001). *Manias, Panics, and Crises: A History of Financial Crises.* John Wiley.

Z. Kurd, T. Kelly, J. McDermid, R. Calinescu, & M. Kwiatkowska (2009). Establishing a Framework for Dynamic Risk Management in Intelligent Aero-Engine Control. In: Buth, B., Rabe, G. and Seyfarth, T. (editors), *Proc. 28th Intl. Conf. Computer Safety, Reliability and Security (SAFECOMP'09)*, vol. 5775 of LNCS, Springer, pp. 324-241.

M. Lewis (2010). *The Big Short: Inside the Doomsday Machine.* Allen Lane.

D. MacKenzie (2008a). *An Engine, Not a Camera: How Financial Models Shape Markets.* MIT Press.

D. MacKenzie *et al.*, editors (2008). *Do Economists Make Markets? On the Performativity of Economics.* Princeton University Press.

D. MacKenzie (2008b). *Material Markets: How Economic Agents are Constructed.* Oxford University Press.

R. Mullane (2006). *Riding Rockets: The Outrageous Tales of a Space-Shuttle Astronaut.* Simon & Schuster.

E. Mumford (1983). *Designing human systems for new technology - The ETHICS method.* Retrieved from http://www.enid.eu-net.com/C1book1.htm

C. Musselle (2010). Insights into the Antigen Sampling Component of the Dendritic Cell Algorithm. In. *Proceedings of the 9th International Conference on Artificial Immune Systems (ICARIS 2010).*

I. Nonaka (1994) A Dynamic Theory of Organizational Knowledge Creation. *Organization Science,* 5(1): 14-37.

I. Nonaka & G. von Krogh (2009) Tacit Knowledge and Knowledge Conversion: Controversy and Advancement in Organizational Knowledge Creation Theory. *Organization Science,* 20(3): 635–652.

L. Northrop *et al.* (2006). *Ultra-Large-Scale Systems: The Software Challenge of the Future.* Technical Report. Carnegie Mellon University Software Engineering Institute.

R.F. Paige, F. Burton & S.M. Poulding (2010). Modelling to Support Decision Making in Complex Environments - an Impossible Challenge? In: *Proc. 3rd Workshop on Non-Functional Properties in Domain-Specific Modelling Languages*, CEUR Proceedings, Oslo, Norway.

C. Perrow (1984). *Normal Accidents: Living with High-Risk Technologies*. New York: Basic Books.

RAEng & BCS (2004). *The Challenges of Complex IT Projects: A report of the working group from the Royal Academy of Engineering and the British Computer Society.* Published by the Royal Academy of Engineering, April 2004.

J. Reason (2008). *The Human Contribution: Unsafe Acts, Accidents, and Heroic Recoveries.* Ashgate.

J. Reich, M. Naylor & C. Runciman (2010). Supercompilation and the Reduceron. In: *Proc. Second International Valentin Turchin Memorial Workshop on Meta-computation in Russia,* pages 159-172.

K. Roberts (1990). Some Characteristics of One Type of High reliability Organization. *Organization Science* **1**(2):160-176.

O. Rogers & D. Cliff (2010). The Effects of Truthfulness in a Computing Resource Options Market. To appear in *Proceedings of the International Conference on Advances in Distributed and Parallel Computing (ADPC 2010)*, November 2010, Singapore.

O. Rogers & D. Cliff (2011). The Effects of Market Demand on Truthfulness in a Computing Resource Options Market. To appear in *Proceedings of the 3rd International Conference on Agents & Artificial Intelligence (ICAART 2011).* January 2011, Rome.

J. Rooksby, (2010). *Social Networking and the Home Office.* Technical Report available from: http://lscits.cs.bris.ac.uk/docs/socialNetworkingHomeOffice.pdf

L.M. Rose, D.S. Kolovos, N. Matragkas, J.R.Williams, R.F. Paige, F.A.C. Polack & K. J. Fernandes (2010). Concordance: An Efficient Framework for Managing Model Consistency. In: *Proc. European Conference on Modelling: Foundations and Applications (ECMFA) 2010,* LNCS 6138, Springer-Verlag, Paris, France.

S. Sloan (1981). *Simulating Terrorism.* University of Oklahoma Press.

I. Sommerville, *et al.* (2010). *The Handbook of Socio-Technical Systems Engineering*. http://archive.cs.st-andrews.ac.uk/STSE-Handbook.

J. F. Sowa & J. A. Zachman (1992). Extending and Formalizing the Framework for Information Systems Architecture. *IBM Systems Journal,* 31(3):590-616.

I. Sriram (2009). SPECI: A Simulation Tool Exploring Cloud-Scale Data Centres. In M. Jaatun, G. Zhao, & C. Rong (eds), *Proc. CloudCom 2009*, LNCS 5931, pp.381-392, Springer Verlag.

I. Sriram, & A. Khajeh-Hosseini (2010). *Research Agenda in Cloud Technologies*. Unpublished technical report. http://arxiv.org/abs/1001.3259

I. Sriram & D. Cliff (2010). Component placement effects on data-centre performance scaling. In: *Proceedings of the 15th IEEE International Conference on the Engineering of Complex Computer Systems (ICECCS 2010).*

M. Stamatelatos *et al.* (2002a). *Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners.* Version 1.1. Available from: www.hq.nasa.gov/office/codeq/doctree/praguide.pdf

M. Stamatelatos *et al.* (2002b). *Fault Tree Handbook with Aerospace Applications.* Version 1.1. Available from: www.hq.nasa.gov/office/codeq/doctree/fthb.pdf

G. Tett (2009). *Fool's Gold: How Unrestrained Greed Corrupted a Dream, Shattered Global Markets, and Unleashed a Catastrophe.* Little, Brown.

D. Vaughan (1997). *The Challenger Launch Decision: Risky Technology, Culture and Deviance at NASA.* University of Chicago Press.

D. Vaughan (2005). On Slippery Slopes, Repeating Negative Patterns, and Learning from Mistakes. In Starbuck, W. & Farjoun, M., editors (2005) *Organization at the Limit: Lessons from the Columbia Disaster.* Wiley Blackwell. Pp. 262-275. Pdf here: http://www.sociology.columbia.edu/pdf-files/vaughan01.pdf

D. Vaughan (2006). NASA Revisited: Theory, Analogy and Public Sociology. *American Journal of Sociology*, **112**(2):353-393. Pdf available from here: http://www.sociology.columbia.edu/pdf-files/nasa.pdf

K. Weick & K Sutcliffe (2007). *Managing the Unexpected.* Jossey Bass.

S. Wolfram (2002). *A New Kind of Science.* Wolfram Media.

Dr Radu Calinescu, University of Aston

Radu Calinescu is a Lecturer in Computing at Aston University, and a part-time Lecturer on the Software Engineering Programme at the University of Oxford, a position that he has held since 2005. He was previously a Senior Researcher on the LSCITS Formal Verification research theme at the University of Oxford, and holds an award-winning DPhil in Computation from the University of Oxford. His recent research on adaptive IT systems and systems of systems won Best Paper Awards at international academic conferences, and the generic development methods and general-purpose software tools he devised as part of this work are currently used by external researchers and practitioners involved in the development of self-managing systems. He has chaired or has been on the program committees of multiple international conferences on autonomic, adaptive and complex computer systems. He is a Senior Member of the IEEE and a member of the Editorial Advisory Board for the International Journal on Advances in Intelligent Systems.

Prof. Dave Cliff, University of Bristol

Dave Cliff is a Professor of Computer Science at the University of Bristol. He has previously held faculty posts at the universities of Sussex and Southampton in the UK, and at the MIT Artificial Intelligence Lab in the USA. He spent 1998-2005 working in industry, initially as a research scientist for Hewlett-Packard Labs near Bristol, and latterly as a director for Deutsche Bank's Foreign Exchange Complex Risk Group, in the City of London. He’s worked as a consultant for a number of companies, mainly in media and the financial markets, and has also acted as a consultant and advisor to the UK Government. A Fellow of the British Computer Society, since 2005 he has been Director of the UK LSCITS Initiative.

Prof. Justin Keen, University of Leeds

Justin Keen is Professor of Health Politics in the Leeds Institute of Health Sciences. His main research interests are in the governance of health care, and in particular the application of systems and network concepts to the organization and delivery of care, and the role of information technologies in health care.

Dr Tim Kelly, University of York

Dr Kelly is a Senior Lecturer within the Department of Computer Science at the University of York. He is also Academic Theme Leader for Dependability within the UK MoD-funded Software Systems Engineering Initiative. His research interests include safety case management, software safety analysis and justification, software architecture safety, certification of adaptive and learning systems, and the dependability of "Systems of Systems". He has supervised a number of research projects in these areas with funding and support from Airbus, BAE SYSTEMS, Data Systems and Solutions, DTI/TSB, EPSRC, ERA Technology, Ministry of Defence, QinetiQ and Rolls-Royce. He has published over 150 papers on high integrity systems development and justification in international journals and conferences.

Prof. Marta Kwiatkowska, University of Oxford

Marta Kwiatkowska is Professor of Computing Systems and Fellow of Trinity College, University of Oxford. Prior to this she held appointments at the Universities of Birmingham, Leicester and the Jagiellonian University in Cracow, Poland.

Marta Kwiatkowska spearheaded the development of probabilistic and quantitative methods in verification on the international scene. The PRISM model checker (www.prismmodelchecker.org) developed under her leadership is the leading software tool in the area, cited 2000 times, and is widely used for research and teaching. Applications of probabilistic model checking have spanned communication and security protocols, dependability analysis, nanotechnology designs, power management and systems biology. Marta Kwiatkowska is a Fellow of the British Computer Society. She is on editorial boards of several journals, including *IEEE Transactions on Software Engineering, Science of Computer Programming* and the Royal Society's *Philosophical Transactions A*. She regularly serves as a member of numerous programme committees and is a founding member of the Steering Committee of the International Conference on Quantitative Evaluation of SysTems (QEST).

Prof. John McDermid, University of York

John McDermid is Professor of Software Engineering at the University of York where he heads a major research group studying high integrity systems. He has worked extensively with industry, particularly in the aerospace sector. He is best known for his work on safety, including the Goal Structuring Notation (GSN) which is now the de *facto* standard for presenting safety arguments. He has given courses on safety and software engineering to more than 120 companies on five continents. He has published six books and over 320 papers. He was elected a Fellow of the Royal Academy of Engineering in 2002. He has extensive experience as a consultant, mainly in the area of safety critical systems and software. He advised the UK Ministry of Defence (MoD) on the development of DS 00-56 Issue 4. He is also a member of the Defence Scientific Advisory Council providing advice to the MoD on their research programmes and on a variety of projects.

Prof. Richard Paige, University of York

Prof. Richard Paige holds a chair in Enterprise Systems at the University of York. He previously held an academic post at York University in Canada. His research focuses on software engineering for large-scale systems, particularly investigating the challenges of abstraction and automation when applied to enterprise-wide systems that have substantial reliability, robustness, safety and security requirements. His recent work has concentrated on the design, development and implementation of domain-specific languages and tools for supporting abstraction while enabling automation in the engineering process. He sits on the steering committees of several conferences in the field of model-driven engineering (TOOLS, ICMT, ECMFA) and is on the editorial boards of the Journal of System Architecture and Software & System Modelling.

Prof. Ian Sommerville, University of St Andrews

Ian Sommerville has been a professor of computer science at St Andrews since 2006 and was previously at Lancaster University. His research interests are primarily in complex systems engineering with a focus on dependability, requirements engineering and socio-technical systems. While at Lancaster, he cooperated with sociologists to study complex computer-based systems with a view to understanding the realities of their use and this has led to a long-term interdisciplinary collaboration. He is convinced that by examining social, organisational and human issues that we can build systems that offer faster 'time to value' after they have been deployed. His goal now is to make socio-technical systems engineering a reality where we use our understanding of socio-technical issues in the development process to create more usable and dependable software systems.