Network-Centric Security - Approaches to Collective Run-Time Adaptation

[See Potential Additions 16Dec2003 @ 1615]

Patrick Beautement

E109, QinetiQ Ltd, Malvern Technology Centre,
St Andrews Road, Malvern, Worcestershire, WR14 3PS, UK
patrick@beautement.com or
pbeautemen@qinetiq.com

Abstract. This paper presents an holistic approach to biologically-inspired security in network-centric domains. The network-centric viewpoint looks at the security challenge in terms of the dynamic interactions between all the entities in the information environment. Critically, it considers the 'run-time' properties of the artefacts, actors and interactions as well as the dynamic influences on those interactions / entities as being the key focus of attention - as opposed to the static, design-time engineering of their elements. This paper examines the value of exploiting the emergent properties of complex systems to influence and ensure the collective, adaptive and secure behaviour of functioning, homeostatic information ecosystems. The paper also identifies some innovative technology contenders for net-centric security and describes a recent series of experiments where they were successfully integrated in the context of military operations. Finally, the paper considers some of the future directions that research might need to take to fulfil the promise of these biologically-inspired security techniques.

1 Introduction

Increasingly, innovators look to nature for inspiration when considering the problems of our increasingly complex and interconnected world - so-called 'biomimetics'. Yet one area seems to have been neglected. Almost without exception the natural world is formed, driven and evolves through the interaction of emergent phenomena which manifest themselves at different levels of abstraction right up to the most rarefied levels of human introspection and beyond. However, in stark contrast, we do not employ the same mechanisms as the natural world in the creation of our devices, systems and human artefacts and so miss out on all the potential benefits that are waiting to be accrued. The exponential growth in the use of communication, mobility and information technology is creating an ever more uncertain, highly interconnected, complex and heterogeneous world. The conventional approaches to the 'design-time' engineering of systems, which rely on the systems being closed, linear, optimised, hierarchical and 'static', do not work on complex systems - which are beyond 'conventional' scientific modelling.

It is curious that we are creating ever more complex software to perform essentially simple tasks. In contrast, nature does the converse, with effective behaviour emerging from simple interactions among 'live-ware'. How is this done? Can the principles involved help us deal with our need for security in our increasingly complex human systems? Clearly, understanding how to wield the phenomenon of emergence[1] as a potential tool is part of meeting this challenge, but what part can other phenomena, such as self-organisation, play and what relevance are the current theories of number, information, complexity and chaos? Indeed, are human intelligence, sociology and culture relevant factors too - if so, how do we employ them?

This paper considers the net-centric viewpoint and identifies the macro behaviours required to support the realisation of this concept. The hypotheses of this paper is that security is not something which is layered over the applications and infrastructure as an afterthought, but is a collective property which emerges from the interaction among the elements which have been deployed. The paper identifies some of the technical challenges of this approach and examines mechanisms which could be employed to ensure the collective, adaptive and secure behaviour of functioning, homeostatic information ecosystems as opposed to the static, design-time engineering of their elements. A set of 'contender technologies' for achieving net-centric security are identified and then the paper describes a recent series of experiments where some of these innovative techniques and technologies have been successfully integrated in the context of military operations. Finally, the paper considers some of the future directions that research might need to take to fulfil the promise of these biologically-inspired security techniques.

2 The Net-Centric Context

So what are the benefits of a net-centric approach and which collective behaviours should net-centric communities display? In the USA, military writers have been mapping out the Network-Centric Warfare[2] (NCW) concept and its implications in some detail [1, 2, 3, 4 and 5]. In the UK, similar concepts are expressed in terms of Network Enabled Capability[3] (NEC). What all these initiatives

[1] Emergent phenomena arise from local interactions among components in an environment, where phenomena persist over time and cannot be deduced by examining the components in their inactive state". See also "Exploiting the Phenomenon of Emergence as a Force Multiplier" at http://www.tbt.org.uk/ and the books "Emergence" by Steven Johnson and "Emergence - From Chaos to Order" by John H Holland.

[2] Network-Centric Warfare, Web site: http://www.dod.mil/nii/NCW

[3] Network Enabled Capability, Web site: http://www.mod.uk/issues/nec

share is a desire to be able to deal with the moment-by-moment realities of modern conflict which is uncertain, dispersed, fluid and where the political and media constraints change moment by moment. The NEC initiative has identified core themes and needs that are particularly pertinent to the security challenge. These have been defined as follows:

- Full Information Availability: enabling a user to search, manipulate and exchange information of different classifications captured by, or available in, all sources internal and external to the organisations involved in a crisis (to aid their cognition)[4];
- Shared Awareness: enabling a shared understanding and interpretation of a situation, and potential courses of action amongst all elements in the crisis (also to aid team cognition);
- Flexible Working: enabling assets to reconfigure rapidly to meet changing mission needs, allowing them to work together with minimum disruption and confusion;
- Agile Mission Groups: enabling dynamic, high-tempo team creation and configuration of groups (using whoever and whatever is available) that share awareness and that co-ordinate and employ a wide range of systems for a specific activity;
- Synchronised Effects: achieving overwhelming effects within and between groups by co-ordinating the most appropriate assets available through dynamic distributed planning and execution;
- Resilient Information Infrastructure: ensuring information resources can be managed and that secure access is robustly assured with the flexibility to meet the needs of agile groups, regardless of adverse communications conditions;
- Fully Networked Support: allowing the ready use of government bodies, industry, academia, public service capabilities and the military to support crisis operations;
- Mobility: allowing context-aware access to information whatever the location of the end user;

As these themes illustrate, increasingly, forces are formed in an ad-hoc manner from 'come-as-you-are' elements. Gone is the 'conflict by numbers' which characterised the Cold War - in its place are new imperatives for command agility and flexibility, with commanders all levels being able to grab the fleeting opportunities which make a difference - without recourse to a cumbersome command chain. In addition, conflict increasingly involves non-government organisations (NGOs) and, most importantly of all, civilian and commercial suppliers - making for a very diverse and highly interconnected environment.

The themes also emphasise that the command process is driven by decision-makers who need to be able to access relevant information as and when they demand it to support the current ‘run time’ imperative and their decision making style [5a, 5b Hutchins]. Information should not be pushed according to some rigid, pre-determined process. A balance is required between formal and informal processes [6 Chin and Clothier]. Indeed, such 'ad-hoc' problem solving is the core task of decision-makers and involves the creative use of information manipulation tools leading to shared awareness, so-called 'interoperability of the mind' and command agility - where the decision-makers are the only thing on the critical path - resulting in 'decision dominance'.

Finally, successful execution of military operations or civilian crisis management situations involves maintaining the cohesiveness and coherence of action / effect of a dynamic (ever changing) mix of heterogeneous and disparate force elements which the unfolding crisis will disrupt. This requires that a continuous, pro-active and agile run-time readjustment process is in place. With this comes the overriding need for security - without compromising the flexibility and agility which are the hallmarks of the net-centric approach. These then are the challenges which we must address - to enable:

- Robustness: the ability to maintain effectiveness across a range of tasks, situations, and conditions;
- Resilience: the ability to recover from or adjust to misfortune, damage, or a destabilising perturbation in the environment;
- Responsiveness: the ability to react to a change in the environment in a timely manner;
- Flexibility: the ability to employ multiple ways to succeed and the capacity to move seamlessly between them;
- Innovation: the ability to do new things and the ability to do old things in new ways; and
- Adaptation: the ability to change work processes and the ability to change the organisation.

3 Analysis and Implications

In accepting the challenge and in considering the network-centric security viewpoint described above, it is immediately obvious that the required collective behaviours are examples of so-called 'quality attributes' - emergent properties which arise from the complex interactions among the elements involved. From this we can state some axioms, dispel some myths and derive some implications. These are discussed below.

3.1 Axioms

It will be axiomatic to this paper that network-centric communities are therefore:

- Complex adaptive systems (CAS);
- Open (unbounded) with distributed, yet highly interconnected, elements;
- Heterogeneous, where fixed standards and procedures cannot be mandated;

[4] This fits with James Hollan and Edwin Hutchins powerful concept of 'distributed cognition' - where cognition does not just occur inside the head, but is a function of context, interactions, environment, artefacts, team members and 'systems' - see references [5a and 5b].

- Uncertain, with ever-changing membership, involving sets of actors, events, artefacts and interactions;
- Diverse, consisting of societies of biological entities and software and hardware set in environments which span realspace, cyberspace and mindspace - and cover activities in the past, present and future.

3.2 Myths

The human race seems obsessed with overcoming challenges, shaping our world and striving for novelty by conceiving of future states and then enacting them with dramatic effect. However, we are still surprised at the many, varied and apparently unexpected outcomes which occur when we transition our schemes from their models into reality. But should we be so surprised? Foremost among our techniques is deterministic modelling based on a Newtonian view of the world. There is a view, however, that determinism is a myth (Prigogine [7]). Indeed, the deterministic experimental conditions of the science laboratory are not a microcosm of the real world - they are atypical of it - Joseph Ford [8] makes the point somewhat whimsically:

> "Unfortunately, non-chaotic systems are as scarce as hen's teeth … algorithmic complexity theory and non-linear dynamics together establish the fact that determinism actually reigns over quite a finite domain; outside this small haven of order [the 'laboratory'] lies a largely uncharted land … where determinism has faded into an ephemeral memory."

So, even if the universe behaves like a machine in the strictest mathematical sense, it can still happen - indeed it is inescapable (as Paul Davies [9] easily proves) - that genuinely new and in-principle unexpected phenomena will occur. The conclusion must be that determinism is a myth and that we need to look beyond classical science and deterministic models and methods to understand, and then be able to harness, emergent phenomena as a positive tool in a net-centric security environment. Hence, except in certain situations and despite protestations to the contrary [10 Brook], we must therefore dispel the myth that systems engineering is the tool of choice to use to produce the secure, net-centric environment we need. With its emphasis on the a-priori definition of requirements and on establishing properties and features of bounded systems of systems at 'design-time', systems engineering is clearly only suitable for engineering some of the foundations of a net-centric environment.

A third myth is that the security agencies and / or the military will not accept systems with behaviour which cannot be exactly determined in advance. The whole NCW / NEC thrust would not exist and be receiving considerable funding and support at the highest levels if this were true.

3.3 Implications

So, the network-centric viewpoint looks at the security challenge in terms of the dynamic interactions between all the entities in the information environment. Critically, it considers the 'run-time' properties of the information artefacts, actors and interactions as well as the dynamic influences on those interactions / entities as being the key focus of attention. It is concerned with the collective, adaptive behaviour of functioning information ecosystems. The first main implication, therefore, is that we cannot employ design-time engineering to provide the security we require. Indeed, research into CAS [11 Pimm] has shown conclusively that it is not enough to assemble a known set of components (even if they once formed a 'fit' community) and try activate them to reproduce previously effective behaviour - this will not work. What is required is that communities must self-organise and climb to fitness, because it is through the dynamic processes of interaction and adaptation (forming a trajectory over time), along with the active maintenance of homeostasis, that the required behaviour emerges - it cannot be packaged at design-time and delivered at run-time. The importance of this insight cannot be over emphasised.

The network-centric viewpoint does not seek to define all the elements and interactions a-priori, but embraces the realities of information ecosystems - that the composition of, and interactions among, information ecosystems are constantly changing - that their elements are heterogeneous; that there are uncertainties and unknowns and that the elements are ever-changing and widely distributed across open environments. The next implication therefore is that the notion of a closed, defined system (inevitably brittle) with a defended boundary is an anathema.

The final, and most compelling, implication which follows is that understanding how to exploit the emergent properties of complex systems as a positive tool is a challenge we should address. The hypotheses of this paper, therefore, is that security is not something which is layered over the applications and infrastructure as an afterthought, but is a collective property which emerges from the interaction among the elements which have been deployed. The paper will attempt to map out how we might go about providing tools to influence the community so that the required collective security behaviour emerges.

4 Addressing the Challenges

In accepting the challenge and in considering the network-centric security viewpoint and its implications described above, it is now apparent that we need run-time science which will provide us with the tools to work actively and effectively with complex adaptive systems. So, where is this science and what are these tools? The truth is that there are very little of either and, more importantly, there are few people with either the interest, application or knowledge in this area - at present.

The first step is to identify a set of approaches which could be employed to model and aid thinking about security in active information ecosystems. It is only then that one can consider the run-time tools, techniques, technologies and strategies (T3S) which are required to influence collective security behaviour so that it meets the requirements laid down above. Fundamentally, research in these areas must be set in the context of providing integrated, interoperable capabilities - a set of disparate academic tools working in closed information environments are not required. I shall start by briefly examining the phenomenon of emergence and will then indicate some of the mechanisms that we may be able to employ it as a positive force.

4.1 Emergent Phenomena

It is not within the scope of this paper to examine the phenomenon of emergence per se. It is discussed, as a concept, in [12 Holland and 13 Johnson], but neither book seeks to explain how emergence could be exploited as a positive tool. For that we need to look to [14 Morowitz, 15 Beautement, 16 Davies and 17 Lewin]. Emergent phenomena may be generalised as having the fundamental characteristic of being tangible or intangible 'patterns' that persist [18 Holland] over time even though the generators of the patterns themselves may be continually changing (viz: an ant foraging party collecting food has an ever-changing membership of ants). The equivalent in the security environment would be that security can continue to be 'robust' even if staff are going on and off duty. In general terms, it is well understood in that emergent phenomena arise in systems with the following characteristics, ie: with components, substrate, interactions and where synergy, antagony and holism etc are at work. Indeed, emergent phenomena can be found in many different circumstances such as: deterministic situations, open systems with non-linear interactions, far-from equilibrium situations, in fact, just about anywhere.

The implication here is that once the ingredients are in place emergent phenomena seem to arise 'spontaneously' (even relentlessly and unavoidably) without anyone having to do anything - but is this true? The consensus is that it is - and that emergence is a considerable force to be reckoned with and that it is something that we usually fail to exploit. My aim here is to show that it is possible to influence emergent phenomena such that they can be mapped to the required security behaviour.

4.2 Features of Emergent Phenomena

What becomes apparent is that there are a number of identifiable features of emergent phenomena which could be used as run-time security tools. These features are discussed below. To assist with the description, I have used an ants' nest as an example, as ants display so-called 'swarm intelligence' - a rich set of adaptive, emergent defence behaviours that we might wish to emulate.

'Substrate'. There is a substrate / context / framework / environment which supports the activities of components (viz: the ants nest, its passages, food stores etc and the surrounding environment in which the ants exist). The substrate may influence the way in which emergent phenomena arise in many ways, eg by shaping interactions (see the discussions on templates and stigmergy below). In the security context this would mean that we must be able to operate in any of the substrates in which security might need to be manifested - be that realspace, cyberspace or thoughtspace - and that 'tools' that we may use include those that enable the direct manipulation of the spaces themselves.

'Components'. There are some components / agents / elements / parts which are either assembled from other components (in a fractal manner) or which function together as a part of some entity. There must be more than one component (viz: the ants in the nest) and the membership of the community is constantly changing. In the security context, therefore, we should seek to provide actors (ie software and hardware elements) which are active, semi-autonomous and adaptive and which can interact and form groupings.

Sensors and Effectors. Every component should have sensor(s) and effectors which enable interaction across their boundaries (see interactions below). At the simplest, this may mean no more than the 'ability' of a water droplet in a standing wave cloud to gain and loose energy and change state. In a more elaborate example, such as the ant's nest, sensing is multimodal and involves an element of 'sensemaking' (to generate some level of internal representation to support computation and decision-making) followed by some action (again multimodal). In our security context, we may employ similarly simple or elaborate sensing and effecting.

Interactions. Interactions exchange information and are necessary if emergent phenomena are to arise. They take place between the components and their artefacts at various levels of complexity and sophistication and are mediated through many types of tangible and intangible mechanisms (in the ant's nest they involve touch, chemical / pheromone messaging between individual ants and the whole nest, individual and collective behaviours, 'crowd' movement, etc). Note that 'structure' may be achieved through communication, eg ants may be physically unconnected but use pheromones to effect communications - they are thus connected in a manner[5]. Interactions also take place between the components via the substrate (so-called stigmergy[6]) and between collections of components in this 'entity' and those in others (ie this ant's nest vs another ant's nest). In the security context there are clear equiva-

[5] It is possible to synchronise the behaviour of chaotic systems by message passing - eg: two pendulums on a wire - so called 'entrainment'.

[6] Where communication occurs by manipulating artefacts in the environment - eg where termites build nest structures but do not 'talk' about nest building. See Di Caro, G., and M. Dorigo. "AntNet: Distributed Stigmergetic Control for Communications Networks." *J. Art. Int. Res.* 9 (1998): 317-365.and Beckers, R., 0. E. Holland, and J.-L. Deneubourg. "From Local Actions to Global Tasks: Stigmergy and Collective Robotics." In *Artificial Life IV*, edited by R. Brooks and P. Maes, 181-189. Cambridge, MA: MIT Press, 1994.

lents, such as putting messages on agent whiteboards, requesting services, adapting to changes in the availability of processing capability, though it is not yet clear how the different types of interaction affect the classes of phenomena that emerge.

Local Rules and Templates. Following from this, in any environment where emergent phenomena are manifested, simple low-level 'local rules' are enacted which determine the nature of the interactions which take place. There are many factors which may impinge on the way that the rules are triggered and executed, though it seems that one of the most important is the substrate - including the notion of a 'template'. In the ants nest, a template may be set by the gradient of pheromone distribution around the queen. The 'rule' might then be "if I am placing earth pellets - move away from a more concentrated pheromone till the density is Y - place the pellet" - the emergent outcome is an appropriately shaped wall. In the security context, rules may by triggered by the presence or absence of malicious code and templates may relate to, for example, gradients of bandwidth availability.

Integration and Activation. Though it seems obvious to say it, emergent phenomena will not arise until we activate all the elements mentioned above and add the dimension of time. What now occurs is that patterns appear, persist over time and have a manifestation which can be detected and acted upon within some context at a higher level of abstraction. However, in reality, a true information ecology can probably never be turned off (cf the Internet). In other words, all we will be doing is adding our components and tool to an existing open 'infosphere' and so we will have to hit the ground running. In the security context, we have the opportunity here to detect new phenomena and apply new security measures which have never existed before.

4.3 Other Factors Relating to Emergent Phenomena

Observer(s) and Context. Clearly, some phenomena will emerge whether or not there are observers present (leaving aside the metaphysical argument here). However, other emergent phenomena are an artefact of the observer [19 Bass] and only have meaning in the substrate, ie the context, of the observer (viz: the perception that the ant's nest is 'angry' if poked with a stick relates to the emotions attributed down to it, from the human social world, by the observer). In a security environment a commander may have useful perceptions about emergent phenomena (or abstractions of them) displayed by the opponent - even if the opponent is unaware that such phenomena are apparent - and different phenomena may be perceived in different contexts.

Lack of Reversibility and the 'Arrow of Time'. Some hold the view that emergent phenomena are not reversible - any cause and effect linkage is one-way, but this is strongly disputed [20 Bricmont]. However, even if we could reverse the 'arrow of time' we would not necessarily see emergent phenomena 'unwind', this is because a differently ordered set of interactions would now take place (in the "poking an ants' nest with a stick" example, the nest would appear to calm down for no reason just before we removed the stick). The insight here is that if the security environment evolves towards an unfavourable state, influencing it back is not just a matter of unwinding - the influence required may be obscure or orthogonal to the phenomena being manifested or may rely on allowing the 'system' to self-organise back to a 'known' attractor.

Lack of Central Control. Emergent phenomena are not dictated in advance or controlled or co-ordinated centrally (top-down), instead they usually arise bottom-up and are observed at a higher-level of abstraction. To alter them, one must generally influence at the bottom - and allow the required behaviour to evolve 'upwards'. However, useful creative tension can be achieved by exploiting an observer's top-down view (at some abstraction) with the bottom-up behaviours - providing a route to exert control.

Lack of Dependence on the Existence of Individual Components. Emergent phenomena will persist despite changes in components of the same 'class' - eg: the generators of the patterns themselves may be continually changing (viz: an ant foraging party has an ever-changing membership of ants, or the water molecules moving through a stationary standing wave are always changing though the standing wave remains). Indeed, components can be added and removed without the whole 'system' being decommissioned. However, what is apparent is that the diversity of the components is important, an homogenous environment is an unstable one. In the security context, this will provide robustness and persistence despite malicious perturbations.

Adaptation. Emergent phenomena which arise without adaptation are like snowflakes - beautiful complex patterns, but they have no function. It seems that really useful emergent phenomena 'grow smarter' over time - in other words the local rules and the nature of the interactions change over time in response to evolutionary pressures. This infers that there is some form of hysteresis and learning - though the learning may not need to be encoded in 'data' - it may be represented by changes in trigger conditions or in the trajectory of patters over time. The implication here is that all our elements, interactions and the substrate itself, should be 'plastic'. A fixed environment is a dead environment (see the discussion on Langton's Lambda parameter below).

Self and Non-Self. As mentioned earlier, CAS are, by default, open environments consisting of many interacting and connected elements. This means that the 'coupling' between elements across the environment makes it difficult (possibly even meaningless) to try and identify 'self' and 'non-self'. Even though one human body seems bounded and has an immune system, it is connected so closely to its environment and its community that identifying where their mutual influences stop is nigh on impossible. The implication for the security environment is profound. Enclaves (so-called compartments) and apparently bounded 'secure' communication facilities can be created - yet in fact their security can be an illusion [21 Lewin]. In a network-centric context, security will have to embrace the reality of the open environment and seize the opportunities that its complexity offers.

State and Persistence. Crucially, emergent phenomena at one level may be viewed as components at a higher-level of abstraction - indeed, emergence provides freedom of action at a higher level which is denied at a lower level. However, these persistent pat-

terns (or trajectories of patterns), must manifest themselves in such a way that mechanisms at a higher level can work on them - identifying them and using them as an invariant - they may then interact and reverberate, leading to further emergent phenomena [22 Holland]. Though these mechanisms are currently poorly understood, it is in this way that sophisticated, high-level security behaviour could be generated from components injected at various levels within the environment.

Entropy vs 'Information' and Increasing Structure and Organisation. Hence, emergent phenomena add to structure in the universe. Despite the Second Law of Thermodynamics stating that entropy always increases towards featureless uniformity, some see (though others disagree [20 Bricmont]) that there is opposite trend at work in the universe - that of increasing structure and organisation at ever higher levels of abstraction manifested through emergent phenomena - the so-called 'optimistic' arrow of time [23 Davies]. This may be being achieved by the fact that there are causation mechanisms at work which would not contradict the Second Law. As Donald MacKay [24] says:

> "…whereas in classical physics the determination of force by force requires a flow of energy, from the standpoint of information theory the determination of form by form requires a flow of information. The two are so different that a flow of information from A to B may require a flow of energy from B to A …"

Levels of Abstraction (the Observer observed). As already mentioned above emergent phenomena may have no 'meaning' [7] at the level at which they are generated. Though this sounds like the beginning of an endlessly infinite regress of no value, it is actually a crucial point to understand if the phenomena of emergence is to be exploited. For example, Popper [25] represents this by his description of "World 1 .. World 3 entities"[8]. This kind of idea might be described as follows:

> That there exist higher levels of emergent abstractions at which a simplified representation of the activities of a lower level can be meaningfully manipulated by an observer *outside the system* (assuming that the observers can adapt to a world view that would let them detect the abstractions, let alone make sense of them).

This is discussed at some length by [26 Hofstadter] and beautifully illustrated by the M C Escher lithograph "Print Gallery" [1956] which shows a young man in a gallery observing a picture which includes himself observing a picture of himself …. we, because we are outside the system (at a different level of abstraction), can observe and reflect on this paradox - possibly itself an emergent phenomena. The "so-what" about this is that if we do not learn how to exploit phenomena (such as that of emergence) then *we may never know about some of the capabilities that are waiting to be used because they exist at a higher level of abstraction, of which we are currently unaware - one that conventional approaches will never reveal to us, but that an opponent could exploit.* Conversely, we may be able to outwit an opponent by reasoning about emergent phenomena displayed by the opponent at an higher level of abstraction - which will be invisible to the opponent (and therefore infinitely secure).

Are There Different Types Of Emergence? There appears to be a view that there may be different types of emergence and that they relate to the way that they arise and to their levels of 'connectedness'. Warren Weaver [27] when writing of Shannon's work on Information Theory [28 Shannon] talked of 'simple systems' (with two or three variables), then a second group which he called 'disorganised complexity' (with millions of variables only tractable through employing statistical mechanics and probability theory) and a third type for which:

> "… Much more important than the mere number of variables is the fact that these variables are all interrelated … These problems, as contrasted with the disorganised situations with which statistics can cope, show the essential features of organisation. We will therefore refer to this group of problems as those of organised complexity"

Other viewpoints on forms of emergence relate to the degree to which the outcome can be predicted in advance. Smuts' 'Holism and Evolution' was the source for one of two 'co-emergent' notions of emergence:

> "Simple evolution or 'unfolding' is not emergence. Emergence is a structuralist property, as in Buckminster Fuller's Geodesic Domes or 'Bucky balls'." [29 Smuts]

and:

> "Thus, within the general framework proposed here, one must distinguish between two different kinds of emergence: A. Deducible or computational emergence. There exists a deductional or computational process or theory 'D' such that an emergent phenomena 'P' (observed in a higher-level system) can be determined by 'D' from the lower-level system. B. Observational emergence. 'P' is an emergent property, but cannot be deduced as in (A) above. [30 Baas]

[7] The phenomena may be real but some of the meaning attributed to a phenomena may be subjective or interpreted in a way which only makes sense to the observer.

[8] Popper describes "World 1" as the physical world and "World 2" as the "world of our conscious experiences" whereas "World 3" is the world of the logical contents of books, libraries, computer memories, and the like.

4.4 Emergence - A 'Theory of Everything'?

Some have argued [13 Johnson, 14 Morowitz, 31 Davies] that self-organisation and the emergent properties of complex adaptive systems may be the active mechanisms in a 'theory-of-everything'. This requires us to developing an acceptable representation of complexity, self-organisation, the phenomena of emergence and their relation to the tangible and intangible world.

Certainly there are phenomena which emerge directly from interaction at the physical level (standing waves etc), but there also appear to be conceptual emergent phenomena observed at the level of human consciousness. It seems possible therefore that there may be a continuum of types of emergent phenomena (depending on their tangibility or intangibility, the 'systems' from which they arose, the mechanisms of interaction / underlying theories at work, the nature of the observer / observation, the level of abstraction or other factors) and that characterising and classifying the types and their differentiating features may just be a time-wasting exercise - or extremely valuable. Currently, this should be a topic for further research.

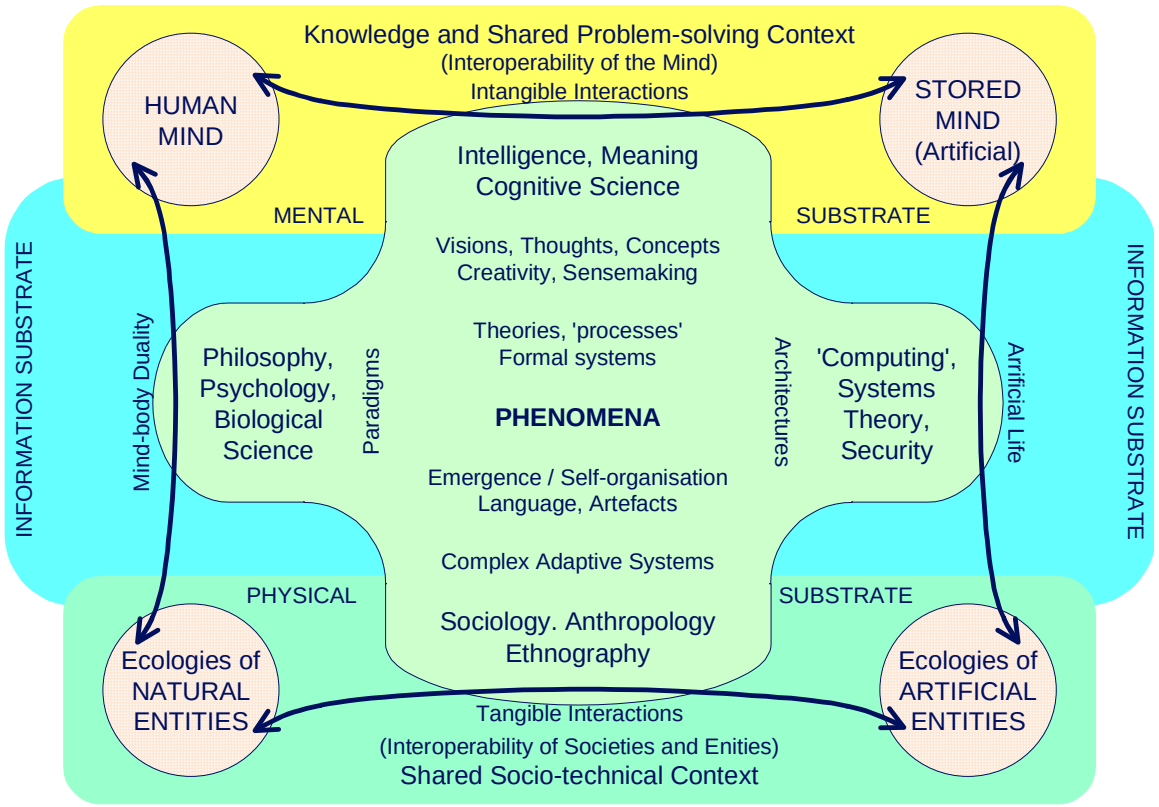

Figure 1 - A Representation of the Domains under Examination

I have made an attempt at this task and offer the diagram at Figure 1 for discussion. It is an attempt to show how emergent phenomena at the physical level may relate to phenomena at the abstract conceptual level, while at the same time showing the relationship between the natural and artificial world. A key part of the diagram is the emphasis on the interactions between the domains shown. It should not be assumed that all interactions are equal - each type of interaction, at each level of abstraction, will be mediated by different phenomena, and security is no different in this respect.

There may well be some 'universal shorthand' for characterising all types of interactions but at this stage it would be foolish to assume this. Indeed, it is clear that it is currently very difficult to 'transform' a viewpoint showing interactions at one level of abstraction into an equivalent one at another level and that this difficulty is a reflection of the incompatibility of the theories and mechanisms underlying the interactions and their attendant representations at each level. Though this seems a long way from where I started, I consider it relevant to try and embrace this broad theme as the notion of security covers, de-facto, the physical, information, mental and social / cultural domains. Indeed, it is necessary to see consciousness as being part of the information processing capability of the biosphere, rather than something unconnected and esoteric. Hence, at some point, we will need to address theories which can span these domains if we are to employ them to provide security 'across the board'.

5 T3S - Likely Contenders

This section of the paper will now discuss the technologies, tools, techniques and strategies (T3S) which could be employed to enable us to exploit the emergent properties of complex systems as a creative force to achieve the required security. From the discussions above, I hope it is evident that we need several types of T3S (see Figure 2) to provide the following:

- Design-time Properties: we need to provide elements which have features which, when activated, display properties that are desirable at run-time - such as being adaptable and plastic - these form our T3S Foundations;
- Tools: we need to have tools and mechanisms available to us at run-time to employ to influence, among other things, the sub-strates and the behaviour of the elements and their interactions - these are the Run-Time T3S;
- Evolutionary Mechanisms: we need to be able to work with, at different levels of abstraction, the self-organising and emergent phenomena themselves to evolve beneficial behaviours (and countermeasures) within the Run-time Environment.

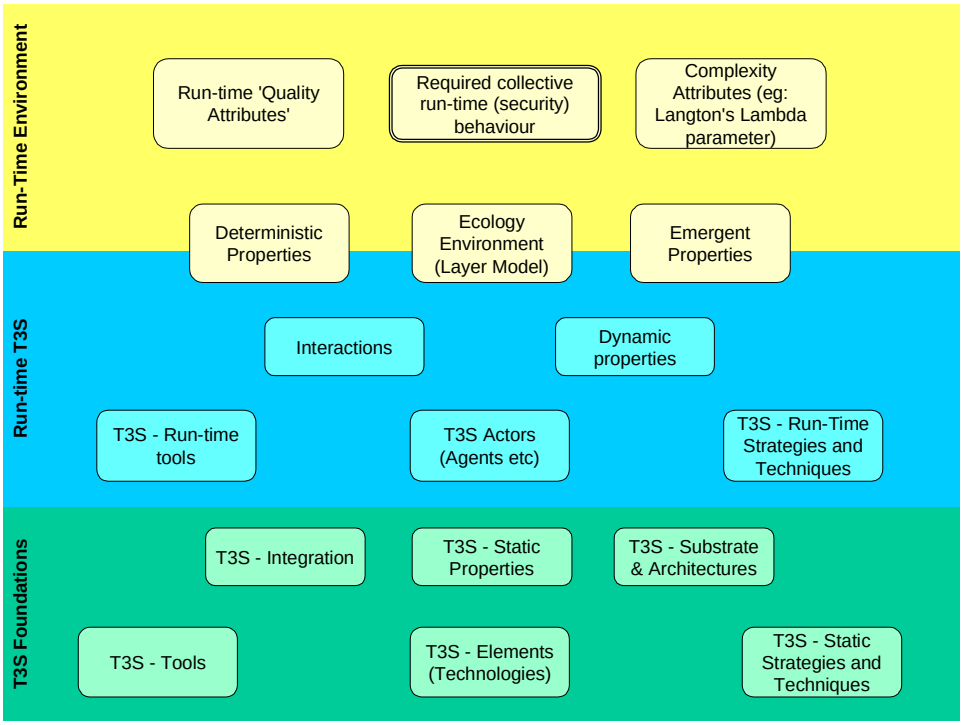

Figure 2 - The Net-centric Environment

5.1 Design-time

In terms of the kinds of T3S Foundations which may be available, over the past few years a number of technologies have begun to arrive that are individually addressing issues of information sharing and access across the distributed communities of virtual or-ganisations. These include agile and resilient networking, peer-to-peer computing (P2P), grid computing, Web services, Semantic Web (knowledge technologies), and human-computer interaction technologies. To enable net-centric security, we will have to harness and intercept the latest technology developments in these areas to enable decision makers to work in a secure, flexible and agile manner. We will need to adopt a number of design principles to ensure that we can meet the requirements demanded for net-centric security. The key design rule that must be applied is to minimise design-time assumptions in order to maximise run-time flexibility. This is an over-arching rule that includes the following principles:

- Separation and Loose Coupling. This includes the separation of interfaces from implementation (separating what a element does from how it does it), and the removal of hidden dependencies and interconnections (using the interface and nothing but the interface). Loose coupling can be achieved using dynamic lookup rather than using predefined connections;
- Exposing Interfaces. As it is not possible to predict how elements will need to be used in the future, systems and components must be designed in an open fashion providing interfaces allowing their functionality to be invoked by new, perhaps unantici-pated elements. This enables reuse and greatly improves interoperability;
- Separation of Action from Information. We should not assume that information is embedded in our active elements and no-where else. The more widely available the information the better the interaction and the greater the flexibility. In security terms, generic data formats and ontologies facilitate this, allowing domain knowledge to be expressed in standard flexible ways, and extracting assumptions, processes or doctrine that would otherwise be hard-coded into devices and thus very difficult to change. These approaches use open standards - strongly supported by off-the-shelf tools - such as wrapping heterogeneous data so that it can be manipulated as shared information.

In creating net-centric implementations today, a number of software T3S[9] exist which have the kind of features we would need and which would, therefore, form a set of 'technology contenders' which could be used for research into exploiting emergence. They are described below in Section 6.

[9] For a more detailed description of T3S Foundations and Run-Time Tools see "T3S for Adaptive Systems" from http://www.tbt.org.uk/

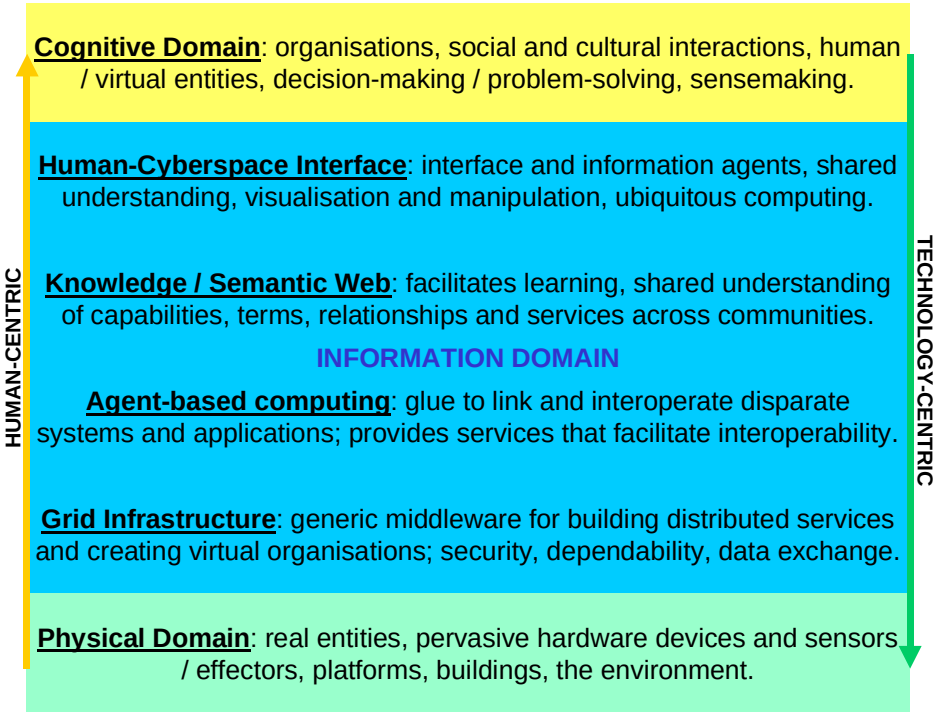

Figure 3 - Layer Model of the Security Environment

5.2 Run-time T3S

Paradoxically, you can't begin to establish the exact performance of components until they interact within the application environment (illustrated at Figure 3) and, even then, all that one can establish is an probabilistic estimate of performance. This is because it is impossible to examine every state under which the components would have to operate as many of the states are emergent and *cannot be part of the formally specified design*. It is appearance of these emergent phenomena which leads to the failure of many of the attempts to create complex system noted above and is why 'unwanted' emergence is treated as something to eradicate and so there is a retreat from complexity towards design-time 'certainty'.

We are turning this view on its head and are looking to maximise novelty at run-time and so accept that we cannot rigorously test the 'system' before deployment because, de-facto, the boundaries of the 'system' cannot be defined. Instead, we will provide ourselves with run-time tools through which we can enforce obligations, allow actions and influence collective behaviour towards our requirements. The real novelty proposed in this paper will come from the innovative integration and application of these T3S capabilities to provide the collective run-time properties that have been discussed earlier.

5.3 Evolving Security through Swarm Intelligence

Though it seems obvious to say it, emergent phenomena will not arise until we activate all the elements mentioned above and add the dimension of time. What now occurs is that patterns appear, persist over time and have a manifestation which can be detected and acted upon within some context at a higher level of abstraction. In the security context, we have the opportunity to detect new phenomena and apply new security measures which have never existed before, but how do we do this, how do evolve the behaviour that we want without trying to impose it top-down? We have already started on the process by altering the design-time features of our T3S Foundations and by creating Run-Time Tools - these will now take effect as they join the active environment. The rest of this section will look briefly at some of the aspects of run-time evolution over which we will have some effect and indicate some of the strategies we could employ and benefits we might accrue.

Collective (Gross) Indicators - Langton's Lambda Parameter. As emergent phenomena arise from interactions (which implies an exchange of 'information') there are optimum conditions under which this occurs and there do seem to be principles at work which would change the nature of the onset and the 'fierceness' of the propagation of the effects. Through his work on cellular automata (CA) and artificial life Chris Langton [39] noticed a pattern to the types of CA which were derived as shown in Figure 4. The pattern also related to Stephen Wolfram's [40] system where he had classified Cellular Automata (CAs) into four classes. Langton noticed the following:

- It became apparent that in situations where the 'information' used by the CAs was constrained to be sparse or 'frozen' few CAs existed - and those that did displayed a narrow 'fixed' behaviour.
- At the other extreme where 'information' moved too quickly to be captured it 'boiled off' and the few CAs here displayed a chaotic behaviour and had hardly any form.

- He found that at a set value (about 0.273, where the complexity of the system was at a maximum, available entropy was at an optimum and where there was a phase change) the most diverse and vibrant CAs were to be found - he dubbed this "life on the edge of chaos".

It appears the phenomena of emergence is also related to this factor and that, in general, CAS evolve and self-organise towards the point of optimum information flow. In principle, this means that we now have a measure which can be applied to the features of a CAS such that some assessment can be made of the likelihood that: nothing interesting will happen, 'interesting things' may happen (eg: at phase transition), the 'system' will generate 'chaotic' emergent phenomena.

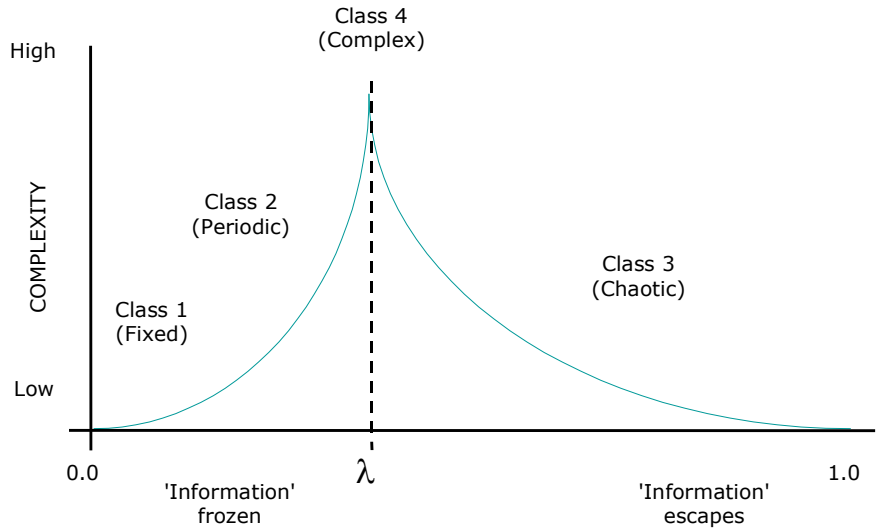

Figure 4 - Langton's Lambda Parameter

Evolutionary Mechanisms. As well as sensing the Collective State (so-called 'swarm intelligence') of the information ecology, there are evolutionary measures which we can employ (at many levels of abstraction) to change the security environment and the specific behaviour manifested which include:

- Fitness and Fitness Landscapes. The entities in the 'infosphere' will all have some level of 'fitness' in relation to the environment and the other components. This fitness is determined by a number of factors such as their degree of generalisation or specialisation, the presence of predators, the number of competitors, the 'richness' of the environment, amount of interactions etc. All these factors can be influenced and, when 'fitness landscapes' are connected, behaviours can be driven towards or way from the current state. A great deal of work has already been done in this area, but it has not yet been employed to influence emergent phenomena in an information environment;
- Population Variation. The balance of members of a population (trophism) is a measure of the diversity and balance within an ecology. Population membership can be altered to trigger a collective change of state to a more beneficial one;
- Downward Causation. Through the connectedness of CAS, reverberation or oscillation can occur where higher-level behaviours affect lower level elements (as with the Laser). This mechanism could be employed to induce a kind of top-down 'control', to impose secure behaviours;
- Environmental Manipulation and Templates - Triggers and Thresholds. Many of the 'local rules' activated by the components in our ecology are triggered by various phenomena in the environment. As has already been mentioned, components may manipulate the environment themselves as a kind of message passing (stigmergy) and we can interfere with this process to alter behaviour. In addition, in an ant's nest, pheromones are used to: leave trails, indicate presence or absence of threat, affect building behaviour (see templates discussed above) etc. This is called chemotaxis. In the security environment, the messages passed between our agents are like pheromones we can alter how long they persist, how quickly stimulation causes fatigue (so messages are ignored) etc. These can be manipulated to effect the required behaviours;
- Interaction Tuning. A variation on this is interaction tuning. We can change the pattern of interaction by altering the frequency and amplitude of messages, apply damping to slow the collective information exchange and move the community from one side to the other of the Lambda Parameter (eg: force opponents into the 'freezing' zone to deny them access to information - eg in virus throttling). In addition, we can alter the 'connectedness' of elements (how much one depends / is affected by another) and so affect the extent to which changes and perturbations propagate through the ecology;
- Learning. Learning is expressed in many ways and may be represented by both static elements (artefacts in the environments) and dynamic elements (reverberating patterns or trajectories of interactions / strange attractors). As learning is part of the way CAS adapt, we can alter the learning artefacts to accelerate adaptation to respond to damage by an opponent or to cause 'forgetfulness' to deceive an opponent;
- Social and Cultural. Lastly, there are also social and cultural effects to be exploited - though these may be at the level of the human decision-makers. Our options here relate to: styles of control (centralised or dispersed), planning and decision-making strategies (a-priori formal or adaptive, continuous and informal) and cultural (autocratic or mob - leading to rapid, chaotic responses). We would currently call this information or psychological operations.

As yet, these approaches are immature and need to be researched and investigated further, yet there is clear potential here, ready to be exploited to help us achieve security in a net-centric environment.

6 An example

The section briefly describes a set of T3S demonstrated in 2002 in a net-centric context at the Coalition Agent eXperiment (CoAX)[10]. The demonstration consisted of a distributed multi-agent system which used the DARPA Control of Agent-based Systems (CoABS) "Grid" to provide the substrate and interaction support. The T3S selected provide capabilities which could be employed by military forces to enable them to dynamically adapt information ecosystems to support the kind of military imperatives which emerge during net-centric conflict. These T3S, which can self-adapt at 'run-time' (within set policy / behaviour controls), augment the cognition of decision-makers and proved superior to 'traditional' approaches which require constant top-down supervision or onerous design-time engineering. This experiment has been recognised[11] as having significant military relevance as a crucial piece of the Network-Centric Warfare puzzle.

CoAX involved the following partners: AIAI, BBN, CMU, Dartmouth, DSTO, GITI, Lockheed Martin ATL, NRL, Potomac Inst., QinetiQ, U.Maryland, U.Michigan, UT-Austin, UWF/IHMC with support from: AFRL, ARL, Boeing, DRDC, DSTL, ISX, Mitre, MIT Sloan, NWDC, OBJS, Schafer, Stanford, TTCP, USC/ISI, US PACOM. The aim of the Coalition Agents eXperiment (CoAX) carried out by this team was to address the unique aspects of achieving coherent Coalition operations from diverse 'come-as-you-are' elements. The operational and technical objectives of CoAX were to:

- show how flexible, timely interaction between different types of (sometimes incompatible) systems and information 'objects' is effectively mediated by agents - leading to agile C2 and improved interoperability;
- show how ease of composition, dynamic reconfiguration and proactive co-ordination of Coalition entities leads to adaptive responses to unexpected events at 'run-time' - providing robustness in the face of uncertainty;
- show how loosely-coupled agent architectures - where behaviours and information are 'exposed' to the community - are more efficient and effective than monolithic programs.
- show how agent policies and domain management help facilitate selective sharing of information between Coalition partners - leading to coherent operations and control of appropriate agent behaviour and interactions - leading to an assured and secure Cyberspace environment.

6.1 The CoAX Demonstration and Scenario

In the earlier part of the demonstration scenario, UN forces had been deployed for some time in a beleaguered state in east Africa and were focusing on the challenging Execution Phase of conflict, on 29th Sep 2012, for which the plans had been produced and the orders issued. The execution of the day's attack missions had been completed and, earlier in the day, media concerns about the location of rare animals in a wildlife refuge (near the attack) caused short-notice replanning to take place with only minutes to spare. New orders were disseminated, plan elements deconflicted and events tracked and handled. Also, the opponents flew hostile air-to-air missions against UN 'high-value assets' and both humans and agents responded. In addition, denial of service attacks had been detected and thwarted by closing down malicious agents at run-time - without knowing anything about their code. By early evening a number of time-critical military events unfolded in the CoAX 2002 Demonstration:

- Part 1 showed how a submarine attack on an Australian ship was reported to the Coalition through the software agent network distributed across the Coalition which adapted the connectivity to support the messaging;
- Part 2 showed the collection and distribution of casualty information - by agents taking mobile code to the affected ship - the outputs triggering automated tools for planning and co-ordinating to align the necessary medical evacuations with ongoing logistics operations - enabling timely rescue and treatment, whilst maintaining patient confidentiality;
- In Part 3, a new country joined the coalition "on-the-fly" (using agents and the CoABS Grid) and agents provided run-time interoperability - enabling the Coalition to use an information feed from an underwater sensor grid provided by the country. National sensitivities were accommodated by employing suitable domains and policies;
- Part 4 showed how agents assisted with the fusion, sharing and employment of the information from the underwater array information along with that from other Coalition anti-submarine warfare forces and sensors so that the enemy submarine (which launched the attack) was detected and its route predicted before it could launch missiles at land-based Coalition forces;
- Part 5 showed how agents helped the Coalition disseminate the information - sanitising sensitive aspects on the fly - so that it was usable by existing Coalition information manipulation tools. This enabled countermeasures to be deployed - resulting in a successful end to the conflict.

[10] For a full set of reports on CoAX and links to information on the DARPA Grid see http://www.aiai.ed.ac.uk/project/coax/demo/2002/

[11] RADm Sprigg, Director USN Warfare Development Centre, 1 Feb 2001 at DARPA CoABS Workshop in Miami.

6.2 CoAX T3S Foundations

CoAX employed a set of T3S foundations which had design-time properties such that useful features could be invoked at run-time. Heterogeneity was embraced, and many standards interoperated. The foundations employed are briefly described below.

Grid Computing. Grid Computing seeks to build infrastructures that enable the integrated collaborative use of high-end computers, networks, databases, and scientific instruments owned and managed by multiple organisations. The term “grid” is used by way of analogy with electrical power grids, where the end-user doesn’t know or care how the power arrives at the socket – they just plug their appliance in. The Grid Computing vision is to provide computing resources in the same transparent way. Although power grids are build from static hardware, the power generated by a particular source can vary wildly - computing grids are even more dynamic in that resources can appear and disappear as well as vary in output. Foster and Kessellman define the Grid as “an enabler for virtual organisations: an infrastructure that enables flexible, secure, co-ordinated resource sharing among dynamic collections of individuals, institutions and resources”. Their book [32] is now widely recognised as the manifesto for Grid computing. Clearly, the Grid could provide part of the substrate required to support net-centric security. In CoAX, the CoABS Grid provided an information substrate in the spirit of the Grid metaphor.

Peer-to-Peer (P2P). P2P is an approach to distributed computing where nodes have significant or total autonomy from central servers. All nodes or peers are equal (though some may be more equal than others). P2P is dynamic; easy to set up, self-organising, with little or no central support required. It copes with environments with unstable connectivity and is inherently robust to attacks as there is no single point of failure. It is possible to provide virtual central control over systems by using policies (see Run-Time Tools below) - if peers co-operate in distributing and enforcing them, i.e. use decentralised functionality and centralised (though asynchronous) control. Legacy systems may be wrapped to allow them to enter a P2P network. P2P is still a relatively immature approach to providing robust communications amongst distributed elements where there is unstable connectivity which makes it ideal as both components and substrate for our net-centric environment. CoAX employed a P2P architecture along with others, enabling flexible interaction between many of the 'come as you are' elements.

Components. Component Oriented Programming can be viewed as taking Object Oriented Programming one step further. A Component is the combination of a work interface, and the implementation of that interface. Its use provides a looser coupling between objects, allowing the implementation to change independently of its clients. A more precise definition: “…a unit of composition with contractually specified interfaces and explicit context dependencies only, which can be deployed independently and is subject to composition by third parties” [33 Szyperski]. Software objects tend to be tightly coupled, and cannot be independently replaced or changed. Components, which may be implemented internally using objects, must be only loosely coupled, with only explicit, declared dependencies. The interface that a component exposes can be advertised, creating a service, which can be discovered and invoked by some, more proactive, entity – an agent, which may be a human user or a software agent (which also might be implemented internally using objects). Agents can be regarded as active components exploring in a marketplace of passive services. Many systems focus on assembly of components by software developers to create a deployed system, but there is also potential to use components to help provide flexibility for systems to assembled and configured by end-users. Web services (see below) can be seen as components combined over the internet. Components are a way to manage complexity, allowing us to create flexible, distributed systems, rather than “stovepiped” monoliths - making them suitable as components of our secure environment - whether used singly or composed into components of components. Many of the active CoAX elements were composed from components, supporting dynamic swapping of capabilities.

Wrapping. The wrapping approach extends the functionality that an existing application or system exposes, leaving the actual application largely or completely unchanged. If the application is to be completely unchanged, additional software can be attached to the interfaces that the application presents to its environment. This can intercept outgoing communications and modify them, and also accept incoming communications in some new form and map them to the form expected by the original application. In other cases, superficial modifications can allow the addition of external interfaces that were not originally present. In either case, the wrapper can add support for new communications protocols. Wrapping is a more general case of top-down componentisation, where the original monolith is kept in one piece, but exposes a number of new external interfaces. In bottom-up componentisation the monolith is split into its sub-systems, each presenting an external interface. Hence, wrapping would enable us to take existing 'non-run-time-aware' security applications and provide them with the kind of interaction capabilities needed for the net-centric approach. In CoAX a number of legacy command and control applications and systems[12] were wrapped and integrated into the dynamic, agent-enabled environment - some were integrated dynamically at run-time in minutes.

Web Services. Web services can be seen as a combination of software component technology and web technology, that allows services from any provider to be discovered, composed and invoked by any consumer, regardless of computer language, operating system or hardware. The key innovation is that there are simple and universally agreed standards, and that web services will be accessible over the web without users being tied to any vendor platform. This is what makes Web Services different from component frameworks such as COM, CORBA, J2EE etc. "Web services" boil down to advertisement, discovery and invocation of functionality over a network, using standard protocols and languages and are closely related to components and agents; there is also convergence with Grid computing. An interesting and important application of Web Services is for multi-channel service delivery (i.e. delivery to heterogeneous devices - phones, PDAs and even 'non-IT' devices such as robots). This is currently done in a stove-

[12] The Master Battle Planner (MBP), the Consolidated Air Mobility Planning System (CAMPS) and elements of the Global Command and Control System for the US Navy (GCCS-N). Details are provided on the CoAX web site.

piped way, with a different application for each delivery platform. Web services are also important for intra- and inter-organisation interoperability. Web services are an immature technology for improved interoperability and service delivery, but with enormous industry support and very rapid improvement expected they could provide extra richness to the interaction mechanisms available within the net-centric environment and enable the generation of security capability on demand. CoAX employed a number of web-services to provide meteorological and intelligence information from the Internet.

Generic Data Formats. Rather than defining a custom, proprietary data format for each application, this approach involves re-using standardised syntax to define the data structures. These can then be labelled using domain-specific terms, but the overall structure is general. These standards can be thought of as "meta-languages": they require the addition of domain-specific terms to create an actual useful language. They are generally known as "mark-up" languages since they often involve adding tags to plain text to make its structure apparent. Standardisation is the fundamental point of this approach, but XML and its relations are not *constraining* standards but *abstraction* standards – they are tools for structuring, not an actual set of fixed definitions. In most applications, the choice of data syntax is largely arbitrary, so it makes sense to choose an existing, standard approach and gain the benefit of existing 3rd party tools, avoiding duplicated effort. We would use these knowledge languages to assist with the representation of shared information and even for structuring internal learning in components, but this will not help us with the issue where transient phenomena (such as reverberating patterns or trajectories of interactions) represent the state of the community. Indeed though, they may not need to represented as such as they are emergent and self-sustaining dynamic phenomena. In this way we could store sensitive information which, because it is not 'located', may by exceptionally secure. CoAX employed XML, RDF and DAML (for a good discussion of the topic see [34 Allsopp]).

6.3 CoAX T3S Run-time Tools

CoAX employed a varied set of run-time tools, technologies and techniques. Some were actors in cyberspace and some provided visualisation and control. The run-time tools employed are briefly described below.

Agents. An agent is a T3S Actor which can be thought of as an active entity situated in some environment that is capable of flexible autonomous action. Agents may act on behalf of, or mediate the actions of other beings and can autonomously carry out tasks to achieve goals or to support the activities of others. An *Avatar* is an agent that represents its owner either in their absence, or at a remote location, and may adopt some human appearance. Conversely, *Interface Agents* represent the system to the user in some way, enabling them to interact with it in some useful way, for example, by eliciting their detailed requirements via a dialogue, delivering information in a user-friendly way or "driving" desktop applications for the user (see Expressive Systems below). *Personal Agents* adapt to their owner's needs and preferences in some way. They may be interface agents, or perform tasks such as managing a user's schedule and arranging meetings in consultation with other people's agents. *Intelligent Agent* is a relative, and sometimes over-used term, but describes the qualities that would make an agent most useful in taking the load off humans – it should be communicative, capable, autonomous, and adaptive, and hence able to carry out complex tasks without constant supervision [35 Hendler]. Apparent intelligence is not purely a property of an individual agent but also of the infrastructure, information sources and other agents in an environment and their interactions, indeed intelligence may be defined conclusively as the ability to create "theories of other minds" [36 Johnson]. Agents provide a powerful metaphor for thinking about and designing distributed and complex systems in a flexible way [37 Jennings]. Agents are particularly suited to dynamic, unpredictable environments where services and connections come and go and where dynamic lookup occurs either via client-server lookup or peer-to-peer discovery. They can be effective at translation and mediation – so-called "glue code" and are closely related to components and web services. Semi-autonomous agents can also be implemented[13] complete with hardware, we would call these robots. As we cannot operate in cyberspace, CoAX employed various types of agents acting on our behalf to help us achieve cyberspace dominance - providing us with an adaptive mechanism suited to the net-centric concept.

Mobile Code. The distribution or migration of software around a network (usually packaged as an agent), to place and execute functionality where it is needed or is most effective is a run-time strategy that we might employ which can be extremely effective in the appropriate niches. Using mobility to save bandwidth and to deal with intermittent connectivity can sometimes be controversial as new advances in mobile communications emerge. However, mobility is also very useful for adding new functionality into specific parts of systems e.g. for ad-hoc training purposes, wrappers, filters and translators. Infrastructure is required to support the hosting and transmission of agents. This is not in itself difficult, but there are many associated security issues when not running on a totally trusted network. "Sandboxing" using permissions policies is a standard approach to protect the host from the agents, as used by Java applets. There are various approaches being researched as to how the agent might be protected from the host. Current security policies may forbid mobility outright as there is insufficient understanding of the issues amongst network security managers. Hence, mobility is an effective solution to specific classes of net-centric security problems, especially where we might need to reach out and insert functionality which is needed at a particular 'location' (eg to provide some certainty in an insecure area), or for network monitoring and analysis. This was done in CoAX to provide surveillance capabilities.

Expressive Systems. Expressive Systems[14] are a run-time strategy which asserts that there should be no mismatch between the objects presented to the user, and the underlying "business objects". Examples of business objects in a security scenario might be *Account*, *User*, *Policy*, *Information Source* and so on. The user can directly manipulate these objects. This requires that the under-

[13] A more detailed description entitled "Constructing Agents - Architectures and Internal Structures" is available from http://www.tbt.org.uk/
[14] See http://www.expressive-systems.org/

lying objects make sense to the end user. A simple example is the computer game "The Incredible Machine" where wheels, ropes, cogs and motors behave as one would expect - enabling them to be easily manipulated to charming effect. In the reference implementation of this concept, known as “Naked Objects”, the user interface is derived directly from the business objects; in one sense there *is* no user interface to “clothe” the objects. Naked Objects makes heavy use of the familiar drag-and-drop idiom, and pop-up menus. Dialog boxes (e.g. “Create new policy? Yes/No/Cancel”) are not used. In a prototype of a booking system for a car company written using Naked Objects there are six types of business object. From these existing instances can be found and new instances created via pop-up menus. A new booking is being created; the date and time are added automatically; the customer and city are added by dragging them onto the booking and a new location is created. These actions can be performed in any order. Objects can only be dragged onto appropriate fields – the user cannot drag a Customer onto a field that expects a Location. The properties of any object (for example, the phone number and current bookings for a particular customer) can be viewed simply by double-clicking on the object. Right-clicking on a booking pops up a menu allowing the user to check availability, confirm the booking, and create a return booking. In the net-centric context, Expressive Systems would provide a security expert with an interface through which they could directly manipulate security issues represented in a manner which fitted with their mental map. For example, a policy could be dragged onto a representation of set of users and implemented automatically. In CoAX, this approach was employed to provide commanders and system administrators with intuitive tools.

Policies and Domains. This run-time tool enables the definition of policies for a system or parts of a system, using a high-level language to describe the desired properties or behaviours of the system. This language is then interpreted and enforced by one or more specific software mechanisms - hence this T3S spans both design-time and run-time. A policy is defined as “an enforceable specification of a constraint on the performance of a machine-monitored action by a subject in a given situation” and usually has a scope which extends over some 'domain'. A domain could relate to the scope of action of an individual, or to some functional area (a military intelligence team) or some arbitrary grouping of agents. Policies give the ability to change behaviour across a domain at different levels of granularity, for example in different sub-domains. Policy fragments or modules can be combined and reused; for example combining a “reduce the quality of data” policy with a “coalition members only” policy would sanitise sensitive information. Contradictory combinations of policies can be detected automatically. Bradshaw et al [38] list a number of advantages of the policy approach. These include: explicit license for autonomous behaviour, reusability, efficiency, extensibility, context-sensitivity, verifiability, support for simple as well as sophisticated systems, protected from buggy or malicious systems, and reasoning about subsystem behaviour. Policies allow implementation-independent specification; specification in understandable terms; modularity of policies; combination, checking and conflict resolution of policies. They are a natural approach to configuring the behaviour of dynamic heterogeneous distributed systems, especially as they can enact global or domain-wide changes without changing code. This would provide security agencies with a mechanism for virtually compartmenting the security environment into domains and then embedding security polices into cyberspace where they will be enacted by agents on our behalf. The required behaviour will then emerge bottom up. These mechanisms are also suitable detecting and counteracting the effects of malicious action by an opponent or their code. All of these features were employed and demonstrated in CoAX.

6.4 Results and Implications

It was felt that the demonstration conclusively showed how an agent-enabled infrastructure could significantly aid the construction and employment of a run-time adaptable Coalition 'command support system'. It was also felt that the CoAX aims were met and that clear benefits had been demonstrated for this prototype net-centric environment. Though few of the run-time evolutionary / swarm intelligence aspects discussed in Section 5.3 were used in CoAX (other than Interaction Tuning and Population Variation), I consider that experiments in exploiting the phenomena of emergence at the conceptual (campaign) level could be devised using the CoAX prototypes as a test-bed for net-centric security issues.

7 Where Next

The hypothesis of this paper has been that security is not something which is layered over the applications and infrastructure as an afterthought, but is a collective property which emerges from the interaction among the elements which have been deployed. However, despite the success of the CoAX experiments, only the tip of the iceberg has been touched - self-organisation and emergence are large topics. Nevertheless, research in this area could lead to the phenomena of emergence being exploited routinely as a useful 'tool'. A multi-disciplinary team would need to be assembled to carry out the following tasks:

- Classify the types of emergent phenomena and their differentiating features and develop a consensus on terminology of emergent properties within the scope;
- Characterise the necessary conditions under which emergent phenomena arise and relate those conditions to the characteristics and classes of phenomena;
- Investigate classes of approach for predicting the emergent properties of interconnecting two or more communities (bottom-up), and investigate classes of approach to predicting what emergent properties are possible given a community (top-down);
- Investigate the use of apparent invariants (such as Langton's Lambda factor and of the Feigenbaum numbers in 'deterministic chaos') to 'measure' factors relating to the generation of emergent phenomena in distributed information systems;
- Investigate the applicability of extensions of classical theories (e.g. to optimum control / optimum filter / organisational management / stability / quantum effects etc);

- Understand how to trigger (and subsequently nurture) certain forms of emergent phenomena 'on demand' to support a specific 'task' and also investigate precursors, constraints and possible 'paths to emergence';
- Research how to exploit the phenomena of emergence in a useful manner in commercial and military applications (mappings between phenomena and their representations in reality);
- Use net-centric warfare (which seeks to actively exploit the phenomena of emergence) as a challenging test case and use the multi-agent systems prototypes available from CoAX as a test-bed for experimentation at the conceptual (campaign) level.

Research Outcomes. The expected outcomes of the research would be as follows:

- Improved understanding of the factors and principles relating to positively exploiting the phenomena of emergence as a 'tool';
- The provision of a set of formal descriptions encompassing characteristics, classifications, invariants and representations relating to the exploitation of the phenomena of emergence;
- The provision of an initial set of tools for employing and exploiting the phenomena of emergence in military and commercial contexts (the 'socio-technical' military context of net-centric conflict and in the commercial context of competition for 'resources' in distributed information systems) with some examples derived from the test-bed research into campaign analysis, military capability management, and information operations with software agent teams;
- Report on the potential applicability of the research information derived from the above to military operations and commercial activities and advice on how problems of this type should be approached.

8 Summary

This paper has examined the concept of net-centricity and its implications for security and has identified clear principles and features which follow from adopting this approach. These include the fact that the network-centric viewpoint looks at the security challenge in terms of the dynamic interactions between all the entities in the information environment. Critically, it considers the 'run-time' properties of the information artefacts, actors and interactions as well as the dynamic influences on those interactions / entities as being the key focus of attention. In other words, it is more concerned with the collective, adaptive behaviour of functioning information ecosystems than the static, design-time engineering of their elements.

The network-centric viewpoint does not seek to define all the elements and interactions a-priori, but embraces the realities of information ecosystems - that the composition of, and interactions among, information ecosystems are constantly changing - that their elements are heterogeneous, that there are uncertainties and unknowns and that the elements are widely distributed across open environments. The notion of a closed, defined system (inevitably brittle) with a defended boundary is an anathema.

What follows from this is the insight that we are dealing with complex adaptive systems where 'conventional' systems engineering provides only a part of the solution. Instead, the hypotheses of this paper has been that in seeking to provide a secure environment, we need to move away from design-time engineering towards run-time evolutionary approaches which exploit the phenomena of self-organisation and emergence. To develop this hypothesis, the paper has attempted to provide some models which would assist when thinking about this problem.

The paper has identified that being able to exploit the phenomenon of emergence in a positive fashion (either as a force multiplier or as a mechanism for effecting change) as key to achieving security in complex adaptive systems. The paper has described some of the features of self-organisation and emergence and has indicated how they could be employed when influencing the secure, collective behaviour of information ecologies. As part of this study, the paper has also short-listed some of the technologies which are contenders for employment when achieving security through run-time evolution. However, this list has not been selected arbitrarily. Instead, it is based on the set of technologies used in the successful Coalition Agents Experiment carried out at Rhode Island in October 2002. As CoAX was, in effect, a testbed of the approaches discussed in the paper, the experiment was described briefly. Finally, the paper considered what further research is required in run-time evolutionary science and what its aims and outcomes might be. The potential is there, it is self-evident all around us - now is the time to exploit it for our own ends.

References

[1] C. Libicki. *The Mesh and The Net*. Martin Center for Advanced Concepts and Technology, Institute for National Strategic Studies, NDU. August 1995.

[2] David S Alberts, John J Garstka, Frederick P Stein. *Network-Centric Warfare: Developing and Leveraging Information Superiority*. CCRP Publications (US DoD Library of Congress). ISBN 1-57906-019-6. 1999.

[3] Alberts, D. S., Garstka, J.J., Hayes, R.E., Signori, D. A. *Understanding Information-Age Warfare*. CCRP Publication Series, 2001. ISBN 1-893723-04-6

[4] Edward A Smith. *Effects-based Operations: applying NCW in Peace, Crisis and War*. CCRP Publication, 2002. ISBN 1-893723-08-9

[5] David S Alberts, Hayes, R.E. *Power to the Edge*. CCRP Publication Series, 2003. ISBN 1-893723-13-5

[5a] James Hollan, Edwin Hutchins and David Kirsh. *Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research*. ACM Transactions on Computer-Human Interaction, 2000

[5b] Edwin Hutchins. *Cognition in the Wild*. MIT Press, Cambridge, MA, 1995

[6] Chin M, Clothier J, Carthigaser M. *Command and Control Capability Assessment and the Criticality Issue*. DSTO. Presented at the 1997 ICCRT Conference. 1997.
[7] Ilya Prigogine. *From Being to Becoming: Time and Complexity in the Physical Sciences.* p214. San Francisco - Freeman. 1980.
[8] Joseph Ford. *How Random is a Coin Toss?* p4. Physics Today. April 1983.
[9] Paul Davies. *The Cosmic Blueprint*. p31. Penguin ISBN 0-14-024362-3. 1995.
[10] Brook, Arnold, Jackson and Stevens. *Systems Engineering - Coping with Complexity*. p94. Prentice Hall ISBN 0-13-095085-8.
[11] S Pimm. *The Balance of Nature*. University of Chicago Press, 1991.
[12] John H Holland. *Emergence: from Chaos to Order*. p7 Oxford University Press. ISBN 0-19-286211-1. 1998.
[13] Steven Johnson *Emergence. The Connected Lives of Ants, Brains, Cities and Software*. Penguin.
[14] Morowitz, Harold J. *The Emergence of Everything : How the World became Complex*. Oxford University Press, 2002
[15] Patrick Beautement. *Exploiting the Phenomenon of Emergence as a Force Multiplier*, at http://www.tbt.org.uk/
[16] Ibid. Paul Davies.
[17] Roger Lewin "Complexity - Life at the Edge of Chaos" Phoenix 1993.
[18] Ibid. John H Holland. p215.
[19] In Nils A. Baas and Claus Emmeche. *On Emergence and Explanation*. Intellectica 1997/2, No.25, pp.67-83 (also as: the SFI Working Paper 97-02-008. Santa Fe Institute, New Mexico, U.S.A). 1997.
[20] J. Bricmont: *Science of Chaos or Chaos in Science?*. In: The Flight from Science and Reason, Annals of the New York Academy of Sciences 775, ed. by P.R. Gross, N. Levitt, M.W. Lewis (The New York Academy of Sciences, New York 1996) pp. 131--175
[21] Ibid. Lewin p210
[22] Ibid. Holland p227
[23] Ibid. Paul Davies. pp20 and 76.
[24] Donald M MacKay. In Nature 232, p679. 1986.
[25] Karl Popper and John Eccles. *The Self and its Brain*. Berlin. Springer International. 1977.
[26] Douglas R Hofstadter. *Godel, Escher, Bach: an Eternal Golden Braid*. p713. Penguin. ISBN 0-14-00.5579-7. 1980.
[27] Claude E. Shannon, Warren Weaver. *Mathematical Theory of Communication*. Publisher: Univ of Illinois Pr (Pro Ref);. (December 1963). ISBN: 0252725484
[28] C Shannon. *A Mathematical Theory of Communication'*. July and October 1948 editions of the Bell System Technical Journal. 1948.
[29] J.C. Smuts. *Holism and Evolution*. MacMillan 1926.
[30] Ibid p5. Nils A. Baas and Claus Emmeche.
[31] Ibid. Paul Davies. p138 onwards
[32] Foster and Kessellman. *The Grid: Blueprint for a New Computing Infrastructure*. August 1998, Morgan-Kaufmann, ISBN 1558604758
[33] Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. 2nd Ed., Addison-Wesley, ISBN 0-201-74572-0, http://www.aw.com/cseng/titles/0-201-74572-0
[34] Allsopp, D.N., Beautement, P., Carson, J. and Kirton, M., *Toward Semantic Interoperability in Agent-based Coalition Command Systems*. Proceedings of the First International Semantic Web Workshop, July 30-31, 2001, Stanford University, CA, USA, pp 209-228.
[35] James Hendler. *Is There an Intelligent Agent in Your Future?*. Nature Web Matters, 11 March 1999, http://www.nature.com/nature/webmatters/agents/agents.html
[36] Ibid. Johnson p195
[37] Jennings, N R, Sycara, K, and Wooldridge, M., *A Roadmap of Agent Research and Development*. Autonomous Agents and Multi-Agent Systems, 1:275-306, 1998.
[38] Bradshaw et al: *Agents for the Masses*. IEEE Journal Mar / Apr 1999. 53-63
[39] Christopher G Langton. *Life at the Edge of Chaos*. On pp 41-49 of A-Life II. Addison Wesley. ISBN 0-201-52571-2. 1992.
[40] Stephen Wolfram. *Cellular Automata as Models of Complexity*. Physica 10D. 1984.

POTENTIAL ADDITIONS

- Cover policies / NOMADS NSA comments from CoAX demos
- Add in cultural control - voting mechanisms as per eBay and /.