PATTERN QUALIFICATIONS AND EXAMPLES OF NEXT-GENERATION AGILE SYSTEM-SECURITY STRATEGIES

Copyright Material IEEE
Paper No. 98

Rick Dove
Member IEEE
Paradigm Shift International
2051 Lama Mountain, Box 289
Questa, NM 87556
USA

Abstract—Current system security strategies are failing and cannot be fixed by security engineers alone. The reason for failure is evident: the attack community operates as an intelligent, multi-agent, self organizing, system-of-systems – with swarm intelligence, tight learning loops, fast evolution, and dedicated intent. Next generation security must engage in true co-evolution, engaging in this arms race with systemic self-organization that leverages community and other forms of multi-agent architectures at least equally agile to the adversary in six observed common characteristics: self organization, adaptable tactics, reactive resilience, evolvable strategies, proactive innovation, and harmonious operations. These concepts cannot be effectively employed by security engineers on sufficient scale without first being enabled by system engineers working at the architectural level. But even then, without appreciation and concurrence by decision makers, self-organizing strategies will fail to gain sufficient deployment. The principal impediment to developing and fielding these strategies is not lack of know-how, but rather lack of a common language and vision that can remove the decision-maker distrust of self organization and unite system engineers and security engineers in architecturally synergistic solutions. This article reports on a cross-discipline pattern project that is discovering and cataloging patterns of self-organizing system-of-systems security. Pattern cataloging projects generally collect best practice history within a single domain. This cross-domain project is necessarily looking into many domains to find recurrent patterns across ecological systems, biological systems, social systems, network systems, enterprise systems, multi-agent systems, ad-hoc networks, unmanned autonomous systems, and others. The intent is to find multiple examples supporting each pattern drawn from disciplines that are comfortable to systems engineers, security engineers, and decision makers—leading to a design and strategy language meaningful to all three. This project began and continues with graduate studies at Stevens Institute of Technology's School of Systems and Enterprises, was adopted as a project activity by the INCOSE System Security Engineering Working Group, and is indicating potential for broader viral spread. This article presents the nature of the project, the qualification filter for candidate patterns, the descriptive form for patterns, selected exemplar patterns, and lessons learned to date.

*Index Terms—s*ecurity, systems of systems, self-organization, security patterns, agility, bow tie architecture., horizontal meme transfer.

I. INTRODUCTION

Operating as self-organizing systems-of-systems, attack communities range from technologically savvy guerrillas and terrorists practicing so-called 4th generation warfare against social infrastructure systems; to system hacker communities empowered by ubiquitous Internet access to tools, techniques, and targets. In the mix are organized crime, entrepreneurial criminals, nation-state war departments, grass-roots flash swarms, and do-it-yourself antisocial expression.

Attack communities are diverse in nature and allegiance, but their strength is rooted in six common characteristics generally absent in current system security strategy: self organization, adaptable tactics, reactive resilience, evolvable strategies, proactive innovation, and harmonious operations. In a word, the adversary is agile.

In recognition of formidable nation-state cyber capability and threat, skill and vast resources play the major role, with self organization in a minor role, e.g, when patriotic grass-roots instigation is employed. Nation-state parity comes through equal attention, and is not of interest to this study as long as traditional forms of attack and defense are the mainstay That is not to make light of the situation, but rather to put this study's focus in context. In time it is expected that nation-states will employ more self organized strategies.

Self-organized strategies are the major characteristic contributing to non-nation-state adversarial success, where top-down organization is not the rule. Two dominant generators of this success are systemically enabled rapid innovation and evolution. The project reported here has begun and will continue to investigate and codify many patterns of self organized security concepts, both reactive and proactive; but this article will single out two patterns directly related to innovation and evolution as core concepts. Both are based on natural systems. One is patterned on horizontal gene transfer in organism evolution, the principal driver behind innovation. The other is patterned on so-called bow-tie architecture, contributing strongly if not centrally to stability in complex systems of many types. Both are featured as important re-usable pattern examples in a later section.

The project reported here got its start with a graduate course on self organizing systems in general, structured as a continuing review of such systems from many domains, and aimed at discovering patterns seen repeatedly across different domains. The objective of the course is to find a core set of patterns in natural systems responsible and necessary for sustainable

IEEE International Carnahan Conference on Security Technology (ICCST), San Jose, CA, USA, 5-8 Oct. 2010, Corrections added 10Nov2010.

effectiveness, that could then inform the design of artificial self-organizing systems. Attack communities, from such as Al Qaeda and other terrorists networks, IED construction networks, various cyber hacking networks, and even do-it-yourself weapons construction communities were among the domains investigated. Notable in this path was the evidence of why traditional security strategy is outclassed consistently, that a security solution strategy would surely have to mine similar characteristics of self-organizing evolution and innovation, and the realization that a security focus might offer a steep ramp for understanding how to build effective self-organizing systems of any kind. Trial under fire, with natural selection driving rapid advancements while engaged in true coevolution with the enemy. This results in a proactive strategic footing with no evident alternative.

The growing body of work in patterns for system architectures has inspired a pattern descriptive approach arising from reviews of Christopher Alexander's seminal construction-architecture pattern work [1], and many others that have adapted the pattern concept to other fields. This project is focused on developing a pattern language that can be common and meaningful to both system engineers and security engineers, and comfortably informative to decision makers.

It is recognized that pattern work is generally a cataloging activity of repetitive proven expressions of architectural strategy (regardless of the domain). There is little in security best practice that stands toe-to-toe with adversarial agility, so this project is an initial path finder activity rather than an organization of commonly known and respected knowledge within the security domain.

How will system engineering facilitate sustainable system functionality in the face of intelligent determined attack? This project's interest is with the security-contextual aspects of system engineering, not with the technology of security.

II. PATTERNS AND PATTERN ORGANIZATION

Security patterns, pattern classification frameworks, and pattern language concepts have had some early attention, but as yet little of it directly addresses self organizing security concepts. Though this project is not yet organizing patterns into classification schemes or a pattern language, that eventual end is informing the selection and development of appropriate patterns.

Perhaps the most ambitious security pattern compendium to date is contained in the 2006 book by Markus Schumacher and others [2]. Though no detailed treatment of self organizing patterns is present, chapter 15 on *Supplementary Concepts* recognizes *Ecosystem-Integra*ted *and Agile Principles* that open the door. An earlier work by Mouratidis, Giorgini, and Schumacher [3] investigates security patterns for multi-agent systems, employing a UML-like representation for depicting how four different agent patterns might relate to each other in collaborative goal accomplishment, an example of pattern language in action. Schumacher and Roedig [4] suggest earlier (2001) that a formal model of a *system* of security patterns would make them "suitable for reasoning and manipulation by software tools", and lists some benefits of design patterns:

- Novices [systems engineers] can act as security experts.
- Security experts can identify, name and discuss both problems and solutions more efficiently.
- Problems are solved in a structured way.
- Dependencies of components can be identified and considered appropriately.

Lacking a grammar, classification schemes are incomplete as a language, but they identify parts of speech, so to speak. Notable attempts include [5, 6, 7] with classification schemes for software security patterns.

Christopher Alexander [1, pp 174-192], the recognized originator of pattern language concepts, suggests that a pattern language function like a grammar, providing sentence structure for combining patterns into meaningful statements. In spoken language there are of course many ways to use any given word and many ways to combine words meaningfully into sentences. Grammar provides fixed rules and modular classifications for assembling a vast quantity of words into an infinite variety of meaningful sentences. As will be seen later, this is a Bow Tie pattern architecture that makes this complexity manageable, while enabling continuous innovation and evolution in language.

Dearden [8] provides a good review of pattern language concepts, not diminished by his application interest for human-computer interaction.

Though not focused on security specifically, Kendall and others [9] in 1997 proposed a seven layer pattern language for agents, addressing how patterns might be organized into comprehensive interactive systems.

Meszaros and Doble [10], reviewing software pattern work, provide a thought provoking proposal for in *Metapatterns: A Pattern Language for Pattern Writing*.

The focus of this article is on self-organizing security patterns, and some significant work has begun in this area. Though not focused on security, Gardelli and others [11] in 2007 explore Design Patterns for Self-Organizing Multiagent Systems. Tom De Wolf's work in self organizing systems is highly relevant. In 2008 he and others [12] reported on *A Catalogue of Decentralised Coordination Mechanisms for Designing Self-Organising Emergent Applications.*

Steven Frank in *The Design of Natural and Artificial Adaptive Systems* [13] recognized the valuable source of self-organizing patterns in natural systems of immunology, evolutionary algorithms, adaptive genetic algorithms, and the learning processes of the brain.

Len Troncale stands out with a first and seminal attempt of a massive and comprehensive compendium of linked natural self-organizing and dynamic system patterns across many different domains of scientific study. His interest is both broad and highly grounded, but digestibly cataloged and condensed into the essence of linked interactive and interdependent processes. His 1978 *Linkage Propositions Between Fifty Principal Systems Concepts* [14] lays the groundwork for his effort, a life work that continues to expand on that original collection and framework, converging on a comprehensive set of transdisciplinary *linkage propositions* between 100 systems processes patterned in his *system of system's processes* [15].

This author is familiar with only one other cross disciplinary pattern project. Though only some relate to security, Dan Lockton with others, in *Design With Intent - 101 Patterns for Influencing Behaviour Through Design* [16], shows the power of cross domain pattern extraction.

III. NEXT GENERATION CHARACTERISTICS

Security is currently locked into a predator-prey coevolution, accepting the role of prey, and reactive in its catch-up actions.

In contrast, next-generation security must at least provide parity with the agility of intelligent attacking systems, observed as six characteristics at the core of attacker agility and seemingly necessary for effective parity.

A. Self-Organization

This is the most important of the six, and is a required system characteristic. It implies a dynamical system composed of components whose relationships reorganize in response to situational forces and events. Reorganization may be caused by willful decision-making agents embedded within the system, by systemic mechanisms that cause seemingly intelligent response, or by a combination of the two. Though decentralized control is favored for robust and innovative reorganizations, centralized reorganizing mechanisms can be effective if sufficiently rule-over-whimsy directed. Order within a self-organizing system is expected, on trend, to increase over time. In practice these systems are in a constant state of self-organization in response to opportunity and threat. If this activity ceases, the system is no longer agile. In a simple sense this causes adaptation. In a more important sense, this is the core of innovation evolution.

The Internet Storm Center [17] is a simple early example of a system composed mainly of independent agents, somewhat transient, both redundant and diverse in functional capability, that wax and wane in population according to the situation at hand.

B. Adaptable Tactics

When an agile system is confronted with a novel situation, it will reorganize its resources in a configuration appropriate to the situation. Adaptability is enabled by an inventory, or immediate acquisition, of appropriate resources. Typically adaptation is what occurs in tactical time frames, and is a real-time response to an opportunity or threat. When the situation allows time for a response, adaptation may include some modification of existing or available resources, provided that the new resource version is compatible with the overall system. Adaptation includes the use of existing available resources in new ways and for new ends.

C. Reactive Resilience

Agile systems live effectively in a world of risk, prepared to recover from disruptive incidents. The term resilience as a systems characteristic has origins in ecological systems, where fires, drought, hurricanes, construction runoff and other such insults disrupt a smoothly functioning ecological system. Ecological resilience allows the absorption of a shock that may alter the affected system for a while, but the system eventually returns to vibrant functionality. The phrase survivable systems is used in computer science, in general conformance with the concept of shock absorption and a possible period of performance degradation, but of course in a much faster time frame.

The immune system is a good role model of a self-organizing, resilient response process: it swings into action when an attack occurs and mounts an aggressive defense. A successful first-time defense learns from experience and is usually able to absorb subsequent attacks of the same nature with little or no performance degradation. Research in the new field of artificial immune systems is advancing quickly and has already resulted in new approaches to intrusion detection products [18].

D. Evolvable Strategies

Evolution takes time to develop new strategic avenues of capability, but don't think of it in biological time frames, slow by "nature." Think rather of John Boyd's OODA loop (observe-orient-decide-act) concept and the need to cycle evolutionary learning loops faster and tighter than the adversary does[19]. Boyd's OODA loop is typically thought of as a tactical concept, akin to competitive adaptability during adversarial engagement, but his fundamental model and the origins of his concept are based on cross-generation evolution of knowledge patterns [8].

Carl Woese [20] has suggested with some forceful arguments that biological evolution was most innovative and rapid in the period that preceded Darwinian evolution, when horizontal gene transfer (HGT) exchanged genes (as components) among single celled entities of different families. HGT works because genes are modular and interoperable with other genes in a cell (system); and because there is an exchange medium for transport of a gene from one system to another. Eventually components within a system become more tightly coupled and more dependent on each other, and the system reaches what Woese calls the Darwinian threshold. This is when the more familiar vertical evolution begins to dominate, and systems become architecturally complex with refinements on stable themes and less able to incorporate new innovations from the outside. We tend to equate evolution with the vertical Darwinian kind, and translate that into automated evolutions based on genetic algorithms. Woese has shown that horizontal evolution will reach an optimum in certain important characteristics and that vertical evolution cannot.

Two implications and one interesting conjecture fall out of this: (1) Adversary communities appear to be evolving toward and through a Darwinian tighter coupled refinement phase, establishing systemic vulnerabilities. (2) Next generation system security strategy might benefit by enabling and leveraging horizontal evolution in order to catch up, and must avoid being seduced by the sizable genetic engineering body of knowledge that practices Darwinian vertical evolution. (3) Adversarial evolution is based on recent ubiquitous connectivity and knowledge exchange offered by the Internet – if security strategy learns to take equal advantage, the asymmetry in current speed and innovation of attacker and attacked should disappear.

E. Proactive Innovation

The term proactive deserves careful attention, since it is often misused. People frequently misuse the term to simply mean active as opposed to passive. One way to know if someone is using the term proactive correctly is to ask whether the person is using it to mean an initiative (one that makes others become reactive) and/or an innovation (something both novel and valuable).

An excellent discourse on proactive security in the face of aggressive engagement is John Boyd's classic, but unpublished, eight-page essay *Destruction and Creation* [19].

F. Harmonious Operation

Embraceable, invisible, synergistic are words that come to mind for describing harmonious security. Usable security is the phrase generally attached to this concept, but sounds weak in comparison.

If a system's security mechanisms are not harmonious with the objectives of the people who use the system, they are not sustainable. Too much of the security effort these days is an imposition on user productivity. The effect is willful user rebellion, with too-frequent disregard and compromise of

security policies, practices, and processes. If security compliance is tough and comes with a personal cost, willful compromise will occur, as well as unintended mistakes.

Natural systems have evolved examples of harmonious security: the immune system, for instance, doesn't maintain a massive infection-fighting population of antibodies on patrol just in case they are needed. Human designed systems have also addressed this need: for instance, making fire-retardant glass every bit as beautiful and transparent as regular glass encourages people to use the fire-retardant type when appropriate. Harmony is a common design principle in construction architecture but not in system architecture—a topic worth exploring at another time.

IV. THE RESEARCH PROJECT

Life adapts and evolves. It is resilient, it is innovative, and it is in harmony with its environment. Lessons from the sustained evolution of species would seem appropriate for consideration in agile security design. Well known are the threat-responsive swarming behaviors exhibited by ants and bees, where large numbers of simple units work together to drive off or eliminate the threat. As individual units they only understand a simple common immediate goal and follow basic rules to achievement. Social animal life exhibits built-in systemic mechanisms for detecting security-threatening behavior among its members, and mitigating behavior if it is evaluated as intolerable. Peer behavior policing is evident in humans, animals, and insect societies.

Without much difficulty early spot-examples of self organized system security can be found. Society evolved police forces only in the last 150 years, and relied on self-organized peer behavior monitoring for most of history. Interestingly, current police theory is moving back toward community policing concepts as being more responsive and custom fit to the neighborhood needs. Ad hoc mesh networks are addressing issues of self policing among nodes. Unmanned autonomous vehicle research is facing issues of peer-peer abnormality detection. Some plants exhibit abilities to detect insect attacks and emit selected gasses that call in the appropriate insect predators. Collective cyber-intrusion incident response communities are emerging among some organizations with similar characteristics. These are only a few domains with appropriate examples for learning.

This project's first phase adopts an initial set of tools that are being evaluated with use and are expected to evolve with experience. The six SAREPH characteristics, so called as the acronym is composed of the first letters of each, mirror key observed characteristics of the attack communities, and are adopted as initial filters for selecting candidate operational patterns of next generation system security.

In the work conducted to date these six SAREPH characteristics are used as filters for selecting candidate agile security techniques. Though no research conclusions guide us yet for suggesting how many or what combinations might be minimally necessary, or even if these six are sufficient, the following guidelines for nominating a candidate pattern are employed:

- It must manifest the self-organizing characteristic and the harmonious characteristic in order to be sustainably agile.
- It must manifest either or both of the evolving and adaptive characteristics.
- It must manifest either or both proactive and reactive characteristics.

Table I shows the current pattern descriptive "form", populated with an early example pattern. The pattern form emerged from an iterative application and discovery activity which identified and described three pattern examples called Dynamic Phalanx Defense, Peer Behavior Monitoring, and Swarming Threat Sensors [21]. All three examples reference multiple supporting instances in the literature in a variety of different system domains. The pattern graphic is a device of the pattern form intended to display time-based response dynamics.

V. PATTERN EXAMPLES

The current mode of pattern description includes a standard pattern descriptive form accompanied with text to clarify the concept. Here the accompanying text is necessarily brief, but should be sufficient to characterize the pattern concept.

A. Pattern: Horizontal Meme Transfer

Evolution is generally known to us in Darwinian terms, where slight mutations and errors in genetic transcription that confer benefit are naturally selected for retention and continued expression. When human-made systems need to evolve automatically, a genetic algorithm is typically considered by designers—where micro aspects of a system are selectively mutated, reassembled into a new system version, and tested for improved fitness, either in simulation or by the real world. Google's so-called split A/B approach [22] to perpetual beta programming is an example of real-world fitness evaluation, in which Google makes daily minor-change releases of developing software and gauges public reaction immediately to either roll back or retain the daily advancing experiment. On a longer time scale, agile software-development processes use a series of iterations with customer evaluations to converge on eventual customer satisfaction.

A recent article in *New Scientist* [23] reviews the thought-shaking work of microbiologist Carl Woese and physicist Nigel Goldenfeld. They have shown that rampant horizontal gene transfer (HGT) predated Darwinian evolution as a necessary precursor, and is responsible for the mysteries of how the genetic code became optimally resilient and universal to all organisms. In Woese's words:

> *Vertically generated and horizontally acquired variation could be viewed as the yin and the yang of the evolutionary process. Vertically generated variation is necessarily highly restricted in character; it amounts to variations on a lineage's existing cellular themes. Horizontal transfer, on the other hand, can call on the diversity of the entire biosphere, molecules and systems that have evolved under all manner of conditions, in a great variety of different cellular environments. Thus, horizontally derived variation is the major, if not the sole, evolutionary source of true innovation*. [20, p. 8393]

Woese's simulations have shown that Darwinian vertical evolution will not find such an optimum, whereas horizontal evolution is driven toward it. As cellular systems evolved more complexity, they eventually crossed what Woese calls the Darwinian threshold, where the preservation and strengthening of internal component dependences becomes favored over the innovative but more risky incorporation of outside components.

How useful it could be to apply this idea to system design? The systems species *Cell Phone* is still in early formative stages of HGT evolution, experimenting with elements of telephones, keyboards, PDAs, World Wide Web, social networking, personal computers, GPS, accelerometers—just to name some starters,

transient as they may be. The well-evolved systems species, *Large Enterprise*, on the other hand, is well known for its fierce maintenance of operating culture, protection of organizational architectures, and rejection of disruptive innovation. It is a well-oiled machine, not about to jeopardize its self identity.

Teams and collaborative groups innovate to the extent of their diversity. This type of innovation is the result of what could well be called horizontal meme transfer (HMT) in action. Common knowledge knows how hard it is to get a new idea accepted in an established "institution."

Woese's concepts explaining evolving systems are consistent with what is seen in the tight-loop learning and rapid innovation of loosely coupled security-attack systems, from Al Qaeda to IEDs to cyber-insecurity. Until we learn how to mirror these *capabilities for systems evolution and innovation, we will remain* behind the security power curve.

Horizontal and vertical system-evolution interplay is a new understanding hidden in plain sight. It has also been discovered by another team from a different angle: highly optimized tolerance, a very HOT idea. Jean Carlson and John Doyle understand something about complex systems and the way they age that provides strong theoretical underpinnings for the behaviors observed in complex systems, ranging from the Internet to the immune system. In their words:

> *Through design and evolution, HOT systems achieve rare structured states which are robust to perturbations they were designed to handle, yet fragile to unexpected perturbations and design flaws. As the sophistication of these systems is increased, engineers encounter a series of tradeoffs between greater productivity or throughput and the possibility of catastrophic failure. Such robustness tradeoffs are central properties of the complex systems which arise in biology and engineering.* [24, p. 2529]

In another paper, Carlson, Doyle and others advance the

TABLE I
PATTERN DESCRIPTION: HORIZONTAL MEME TRANSFER

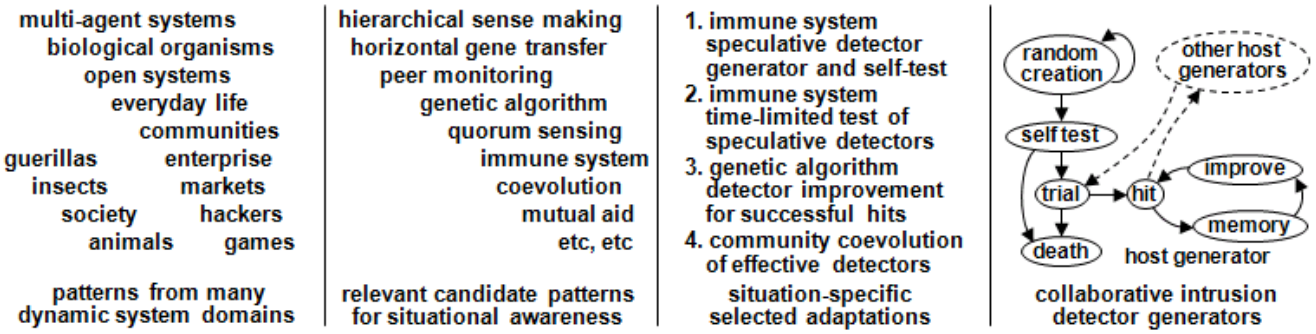
Name: Horizontal Meme Transfer (adapting patterns from other domains)

Context: Systemic innovation and evolution.

Problem: A need for improved system survivability, either reactive, proactive, or both.

Forces: Evolution of innovation vs. evolution of robustness.

Solution: Find relevant patterns in other domains and adapt them to the perceived threats and opportunities of the system of interest.



Example: Massive shared generation of intrusion detectors for evolving resilient-network vigilance [25]

Example: Horizontal gene transfer and evolution. See (Woese 2000) and (Smets 2005).
Example: An example of a cross-domain user-behavior-channeling pattern catalog. See (Lockton 2009, 2010)
Example: Seminal cross-domain dynamic-system process-pattern project, See (Troncale 1978, 2006)
Example: Universal patterns in human activity and insurgent events. See (Bohorquez 2009).
Example: Patterns in behavioral ecology and anti-predator behavior. See (Blumstein 2010).
Example: Tradeoff between robustness and fragility in evolving complex systems.

Agility: Self organization controls the assembly process. Adaptation occurs in assemblies that meet needs. Reactive resilience occurs with sufficient module mix to meet specific needs. Evolution occurs in module and protocol upgrades. Proactive innovation occurs with speculative assemblies for unknown needs. Harmony is maintained with a Highly Optimized Tolerance (Carlson 2002) small module and protocol repertoire in the knot. [S-A-R-E-P-H]

References: (see reference section, only URLs shown here. All accessed 4Jul2010))
(Blumstein 2010) www.eeb.ucla.edu/Faculty/Blumstein/pdf%20reprints/Blumstein_2010_BE.pdf [26]
(Bohorquez 2009) www.nature.com/nature/journal/v462/n7275/full/nature08631.html [27]
(Carlson and Doyle 2000) www.pnas.org/content/99/suppl.1/2538.full.pdf+html [24]
(Lockton 2009) http://bura.brunel.ac.uk/bitstream/2438/3664/1/Lockton_SI_paper_disclaimer_added.pdf [16]
(Lockton 2010) http://danlockton.com/dwi/Download_the_cards [28]
(Smets 2005) www.nature.com/nrmicro/journal/v3/n9/pdf/nrmicro1253.pdf [29]
(Troncale 1978) www.allbookstores.com/author/International_Conference_On_Applied_General_Systems_Research_State_Uni.html [14]
(Troncale 2006) http://www3.interscience.wiley.com/journal/112635373/abstract?CRETRY=1&SRETRY=0 [15]
(Woese 2000) www.ncbi.nlm.nih.gov/pmc/articles/PMC26958/pdf/pq008392.pdf [20]

claim that the concept of highly optimized tolerance effectively explains emergence in complex network systems, unlike any other attempts, which "can be convincingly debunked and are easy to reject" [30, p. 278]—strong words, effectively argued.

Adding robustness initially or incrementally over time creates complexity within the system, preserving and protecting its essential functions and capabilities against known uncertainties. But at the same time, the system becomes increasingly fragile to unexpected threats and so-called Black Swans—unavoidably.

There is small utility in just letting these insights explain the world around us: they should be put to work in purposeful design—this would be horizontal meme transfer in action.

B. Pattern: Bow Tie Processor

At the forefront of systems engineering are adaptable systems that can coevolve in uncertain environments exhibiting unexpected events. It seems obvious that they are needed in security and for resilient networks [25]. But they are also needed to keep up with rapidly changing system requirements, to prolong the life cycles of deployed systems, to accomplish true quick reaction capability, and to develop next generation standards concepts [31] that evolve at the speed of need.

TABLE II
PATTERN DESCRIPTION: BOW TIE PROCESSOR

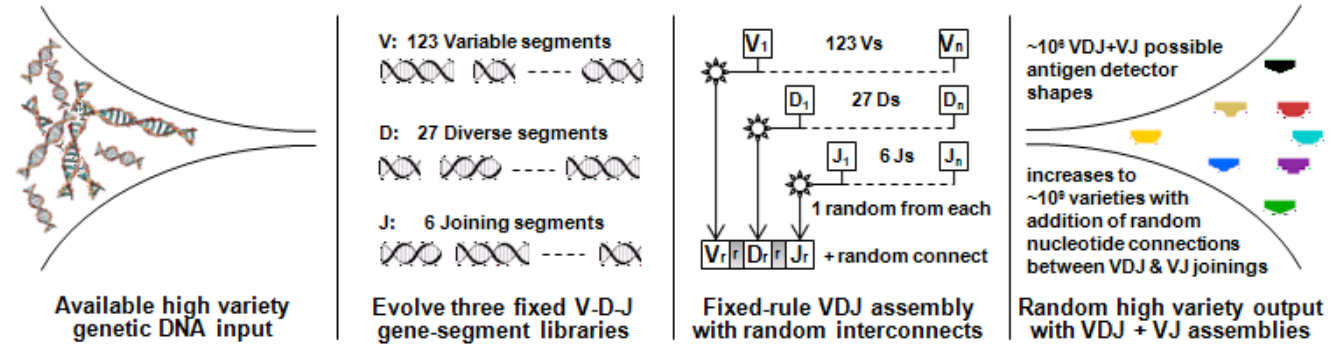
Name: Bow Tie Processor (assembler, generator, mediator)

Context: Complex system with many diverse inputs and many diverse outputs, where outputs need to respond to many needs or innovate for many or unknown opportunities, and it is not practical to build unique one-to-one connections between inputs and outputs. Appropriate examples include common financial currencies that mediate between producers and consumers, the adaptable biological immune system that produces proactive infection detectors from a wealth of genetic material, and the Internet protocol stack that connects diverse message sources to diverse message sinks..

Problem: Too many connection possibilities between available inputs and useful outputs to build unique robust, evolving satisfaction processes between each.

Forces: Large knot short-term-flexibility vs. small knot short-term-controllability and long-term-evolvability (Csete 2004); robustness to known needs vs. fragility to unknown needs (Carlson 2002).

Solution: Construct a relatively small "knot" with fixed modules constructed from selected inputs, that can be assembled into outputs as needed according to a fixed protocol (rules). A proactive example is the adaptable immune system that constructs large quantities of random detectors (antigen epitopes) for unknown attacks and infections. A reactive example is a manufacturing line that constructs products for customers demanding custom capabilities.



Example: Immune system--Millions of random infection detectors are generated continuously by fixed rules and modules

Example: For digestible description of immune system assembly process see (Wikipedia 2010). For numbers see (Li 2004).
Example: Bow tie architecture for detector generation and sense-making. See (Dove 2010).
Example: Bow tie architecture for robust complex networks of many kinds. See (Csete 2004).
Example: General bow tie architecture and flexible-standards generation. See (Hartzog 2010).

Agility: Self organization controls the assembly process. Adaptation occurs in assemblies that meet needs. Reactive resilience occurs with sufficient module mix to meet specific needs. Evolution occurs in module and protocol upgrades. Proactive innovation occurs with speculative assemblies for unknown needs. Harmony is maintained with a Highly Optimized Tolerance (Carlson 2002) small module and protocol repertoire in the knot. [S-A-R-E-P-H]

References: (see reference section, only URLs shown here. All accessed 31July2010)
(Carlson 2002) gabriel.physics.ucsb.edu/~complex/pubs/hot2.pdf [30]
(Csete 2004) www.cds.caltech.edu/~doyle/CmplxNets/Trends.pdf [32]
(Dove 2010) www.parshift.com/Files/PsiDocs/PatternsForResilientNetworks.pdf [25]
(Hartzog 2010) http://blog.p2pfoundation.net/how-different-is-your-bow-tie/2010/06/21 [31]
(Li 2004) http://bloodjournal.hematologylibrary.org/cgi/reprint/103/12/4602.pdf [33]
(Wikipedia 2010) http://en.wikipedia.org/wiki/V(D)J_recombination

The biological immune system is an excellent model. In Steven Frank's words :

...the genetic system spawned an adaptive subsystem to handle the unpredictable challenges of parasitic invasion. ... The challenge … is clearly defined. The response requires recognition of invaders. Adaptive immunity uses a number of techniques to adjust exploration for better recognition of invaders versus exploitation of existing recognition tools. This dynamic balancing between exploration and exploitation occurs on short time scales. ... other adaptive systems rarely provide such clear challenge-response couples [13, p. 455-456].

Fundamentally the architecture has a relatively small library of fixed modules (at the central knot), which are selected from a large variety of inputs (left bow), and that can be assembled into a very large variety of outputs (right bow), according to a set of protocols (assembly rules). The modules and the assembly protocols are relatively fixed and slow to change.

There is no fitness test within the bow tie architecture itself. The output might be viewed as speculative innovations or custom responses, in search of verification. The assembly process is highly constrained according to the protocol rules.

The knot itself is the fragile portion of the bow tie architecture [34], as its modules and rules are relatively fixed, and if they do not succeed in generating outputs that are appropriate for changing needs, the system fails to generate a successful response. The immune system as a whole has other mechanisms that test the fitness of the output, but these tests do not affect the content of the knot, except on biological evolutionary time frames among a large population of similar but individually variable organisms – outside the scope of any one bow tie.

Nevertheless, life as we know it is testament to the effectiveness, based on the evolved selection of modules and protocol. For man made systems, this is where the systems engineering architect must focus creative thought for successful adaptive systems [30].

VI. ALTERNATIVE GRAPHIC DEPICTION

The multi-panel graphic depicted in the two pattern descriptive forms in this article is also used in depicting the three patterns of [21] and the two patterns of [25]. These pictorial representation are experiments at the moment thought to convey potentially memorable visual images. A more general graphic that may capture the key architectural concepts of agile self-organizing systems and systems-of-systems is employed in the graduate courses mentioned earlier and is described in [35]. It has the advantage of capturing additional key and necessary detail, but does so with a consistent pictorial representation that only changes in the labeled elements. Thus, desired graphic distinction is traded off against the illumination of critical implementation elements.

Fig. 1 shows the Bow Tie pattern of Table II represented in this alternate depiction. Note that the modules and assembly rules are clearly itemized, and more importantly, the four responsibilities of sustainable integrity is indicated directly. Implementation guided by such a depiction cannot ignore these critical elements. This author is not a biologist and the lack of a designated repair mechanism is a reflection of a current

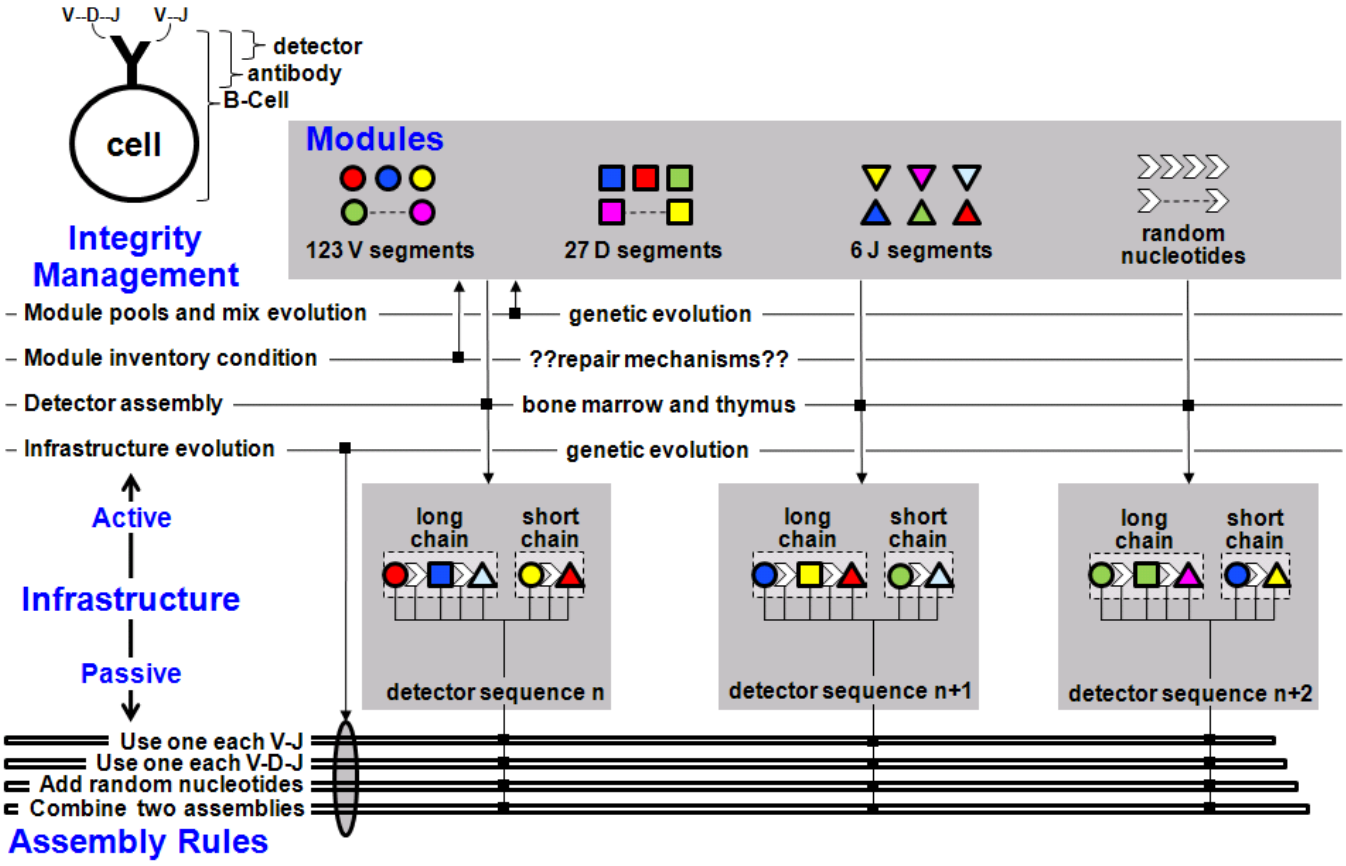


Fig. 1 Adaptable Immune System – Bow Tie Antigen-Detector Generator

knowledge gap. It may be that the immune system has no such real-time repair mechanism, and may rely on a biological evolution-timescale, but an artificial immune system implementation must address this responsibility or decide to live with a glaring known systemic vulnerability.

A principal objective for the reported project is transparent communication, and the multi-panel cartoon appears to accomplish that in a more beneficial way. Nevertheless, this alternate approach would seem to fill a necessary end in pattern implementation understanding. This alternate depiction has been developed as a pattern in its own right, referred to as *Plug and Play* in [36]. A four panel representation of this Plug and Play pattern might consider representing the four integrity management responsibilities. This approach may prove to be a possible universal four-panel depiction structure for all self-organizing systems-of-systems, and needs more thought development

VII. DISCUSSION AND LESSONS BEING LEARNED

This project is in its early stages and feeling its way toward eventual maturity. Several lessons have already made themselves evident.

Graphic Depiction - It is thought that graphics play an important role in conveying and recalling the essence of a pattern. The objective is communication, to decision makers as well as engineers. SysML has been experimented with by some project pattern developers, a natural path for engineers familiar with this modeling language. It was learned that SysML depictions, though graphic, are static and fail to convey the system dynamics. It was also learned that a SysML depiction requires interpretation and reasoned thought, rather than employing the immediate pattern memory and recall of the dominant visual cortex. Flow charts were also attempted by some pattern developers – and though conveying dynamic flow, reasoning and thoughtful interpretation is again required. The HMT graphic in Table I appears wordy and flow-chart-like, yet seems to walk an acceptable middle ground in conveying dynamic concept more akin to the cartoon panels of the Table II Bow Tie graphic.

Graphic Topic - The choice of panel graphic subject is an unresolved concern. Should it be a general pattern abstraction, or an example of a real but specific system? One development effort depicting a Plug and Play pattern [36] made this distinction clear by showing both to good effect. The counter force to showing both is the desire to convey a complete pattern descriptive form on a single page. The depiction of an example rather then a generic pattern is thought by this author currently to be more memorable with its tie to a real system. Necessary accompanying text, more verbose than offered in this article, might well include the generic abstraction. This dilemma remains unresolved. Currently the one-page pattern descriptive form relies on the reader chasing the references to reveal the source understandings of general-pattern abstraction.

Accompanying Text - Text accompanying a single-page descriptive form appears to be necessary. Most pattern and pattern language efforts have codified established best practices in a single domain. This project is working cross domain, extracting patterns from fields of depth not necessarily in common knowledge – the immune system for instance. With maturity a multi-page text accompaniment is envisioned, where a variety of supporting examples from different domains are discussed in brief but sufficient detail to communicate, justify and support the claimed general abstraction. It is also likely that a discussion of the forces and the SAREPH manifestations in each example need to be elucidated, the one to guide application considerations, the other to illuminate holistic integration.

Application Suggestions – It is one thing to read and appreciate a new pattern insight, such as Horizontal Meme Transfer or Bow Tie Processor, but another to recognize where they might be usefully applied. Accompanying text might well suggest a variety of general application areas to open up new lines of thought and kick-start serious consideration.

Speaking to the audience – Ultimately the intent is to convey concepts that can become comfortable to systems engineers, security engineers, and decision makers, each with a different world view, both collectively and individually. Consideration might be given to requiring that examples include at least one likely familiar to each of the three audience categories. A good job of grounding the Bow Tie pattern in complex natural systems is done by [32] with examples of the TCP/IP Internet protocol stack, electric power grid, and financial currencies that mediate trade between produces and consumers.

Pattern Language – Organization of patterns into a pattern language awaits a sufficient number of patterns to suggest or perhaps revel a relevant and useful approach. This objective has been in mind throughout, and the literature cited earlier has offered inconclusive food for thought as yet. The current preference is to provide a grammar for construction rather than a cataloged classification schema.

VIII. CONCLUSION

The need to open meaningful collaborative communication between security engineers and systems engineers seems obvious, but an equal if not primary target are the gatekeepers of system acquisition, funding, approval ,acceptance, and research. These are the decision makers that shape security priority, strategy, and requirements, often passively to decisive effect. Those who develop procurement and acquisition specifications and research priorities are obvious, but the largest and perhaps most decisive communications gap is with CFOs, CEOs, and even some CIOs CSOs focused on organizational priorities that come in front of security. They need to be comfortable with their decisions, but often the engineering language, culture and knowledge barriers inhibit sufficient respect and understanding.

Decisions are rooted in tradeoffs, in three separate silos working within their own areas of control and comfort. This gap is recognized by many of the engineers involved, and relationships have reached a tolerated working balance. But next generation security concepts raise new barriers – they will by necessity be non-deterministic. Decision makers are perhaps the most uncomfortable here where security is concerned. A paradox on two fronts: traditional security demonstrates daily its non-deterministic reality, and decision makers live with comfort in an inherently non-deterministic organizational decision space.

The agility of the adversary and the urgency for effective systemic response offers the systems engineering community a tangible and urgent target for meaningful application. The adversary does not enjoy an engineered system supported by an engineering community, but rather fell into their practices as a natural emergence of a loosely coupled common-interest community. The potential exists for more than parity, with engineered systems that evolve and innovate in advance of the adversary, putting them into the reactive mode.

This project is currently a volunteer activity, principally among members of the INCOSE working group on Systems Security Engineering, and is unlikely to develop the necessary full head of steam until a more formal and funded activity is arranged. Funding aside, a moderated wiki approach is under consideration regardless, but will wait until the basic project platform matures a bit more to avoid wiki chaos.

Recognizing patterns in security strategies, identifying those that are agile, communicating them effectively to decision makers as well as engineers, and defining them in a reusable pattern format can help accelerate the inclusion of agile security as part of the systems engineering process.

This work is developing a population of "path finder" patterns of agile security, expecting they will be replaced with a more historical-based pattern compendium once sufficient experience is accumulated on the way toward a *mature* pattern language. This initial work should provide a platform for subsequent investigation and augmentation.

Much work remains in pattern discovery, description, and organization before a comprehensive pattern language will emerge, but even early beginnings can provide useful building blocks that do not need to wait for the historical perspective before application.

As an example, building on the two patterns shown in this article, plus two more appropriate to situational awareness and sense making [25], and a new pattern-matching processor [37] enabling affordable massive pattern recognition, separate work is investigating feasibility and development for immediate resilient network application.

In closing, the innovation needed in security strategy being pursued by the project reported here is itself based on a core of Horizontal Meme Transfer.

IX. REFERENCES

[1] Christopher Alexander, *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, 1977.

[2] Markus Schumacher, Eduardo Fernandez-Buglioni, Duane Hybertson, Frank Buschmann, Peter Sommerlad, *Security Patterns – Integrating Security and Systems Engineering,* John Wiley & Sons, 2006.

[3] Haralambos Mouratidis, Paolo Giorgini, Markus Schumacherc, "Security Patterns for Agent Systems," Proceedings of the Eight European Conference on Pattern Languages of Programs, EuroPLoP, Irsee, Germany, 25-29 June 2003.

[4] Markus Schumacher and Utz Roedig, "Security Engineering with Patterns," PLoP 2001, Monticello, Illinois, 11-15 September 2001.

[5] Munawar Hafiz and Ralph E. Johnson, *Security Patterns and Their Classification Schemes*, Technical Report for Microsoft's Patterns and Practices Group, September, 2006. Available 31July2010 at https://netfiles.uiuc.edu/mhafiz/www/research/patterns/secpatclassify.pdf.

[6] Munwar Hafiz, Paul Adamczyk, and Ralph E. Johnson, "Towards an Organization of Security Patterns," IEEE Software Special Issue on Software Patterns, 24(4):52-60, 2007. https://netfiles.uiuc.edu/mhafiz/www/research/patterns/haj07-security-patterns.pdf.

[7] Yoshioka Nobukazu, Hironi Washizaki, and Katsihisa Maruyama, "A Survey on Security Patterns," Progress in Informatics, No 5, pp 35-47, 2008.

[8] Andy Dearden, "Pattern Languages in HCI: A Critical Review," Human Computer Interaction, 21(1), January 2006.

[9] Elizabeth A. Kendall, Margaret T. Malkoum, and C. Harvey Jiang, "The Layered Agent Pattern Language," In PLoP '97 (Pattern Languages of Programs), Monticello, Illinois, 3-6 September 1997.

[10] Gerard Meszaros and Jim Doble, "Metapatterns: A Pattern Language for Pattern Writing," In: The 3rd Pattern Languages of Programming Conference, Monticello, Illinois, 4-6 September 1996 .

[11] Luca Gardelli, Mirko Virol and, Andrea Omicini, "Design Patterns for Self-Organizing Multiagent Systems," Proceedings 2nd International Workshop on Engineering Emergence in Decentralised Autonomic System, ICAC, Jacksonville, Florida, 2007, pp 62—71.

[12] Tom De Wolf and Tom Holvoet, *A Catalogue of Decentralised Coordination Mechanisms for Designing Self-Organising Emergent Applications*, Katholieke Universiteit Leuven, Department of Computer Science, Report CW458, August, 2006.

[13] Steven Frank, "The Design of Natural and Artificial Adaptive Systems," Chapter in Adaptation, M. R. Rose and G. V. Lauder, eds., Academic Press, pp 451-505, 1996. Available 31July2010 at http://stevefrank.org/reprints-pdf/96DesignAdapt.pdf.

[14] L. Troncale, "Linkage Propositions Between Fifty Principal Systems Concepts," in Applied General Systems Research: Recent Developments and Trends : N.A.T.O. Conference Series II, Systems Science, G. J. Klir, (Ed.), Plenum Press, pp 29-52, 1978.

[15] L. Troncale, "Towards A Science of Systems," Systems Research and Behavioral Science, Special Journal Edition on J.G. Miller, Founding Editor (G.A. Swanson, Ed.) 23(3): 301-321, 2006.

[16] Dan Lockton with Davis Harrison and Neville A. Stanton, *Design With Intent - 101 Patterns for Influencing Behaviour Through Design*, Equifine, April, 2010. Available 31July2010 at http://www.danlockton.com/dwi/Download_the_cards.

[17] Internet Storm Center. http://isc.sans.org/about.php.

[18] Stephanie Forrest and Steven Hofmeyr, "Engineering an Immune System," *Graft* 4(5):5-9, 2001.

[19] John Boyd, Destruction and Creation, unpublished paper available 31July2010 at www.scribd.com/doc/12627002/Destruction-and-Creation-by-John-Boyd , 1976.

[20] Carl Woese, "Interpreting the Universal Phylogenetic Tree," PNAS 97 (15): 8392-8396, 18 July 2000.

[21] Rick Dove and Laura Shirey, "On Discovery and Display of Agile Security Patterns," Conference on System Engineering Research, Hoboken, NJ, March 17-19, 2010. www.parshift.com/Files/PsiDocs/Pap100317Cser-OnDiscoveryAndDisplayOfAgileSecurityPatterns.pdf

[22] S. Shankland, "We're all guinea pigs in Google's search experiment," *CNET News*, 29 May 2008. Available 31July2010 at http://news.cnet.com/8301-10784_3-9954972-7.html.

[23] M. Buchanan, "Horizontal and Vertical: The Evolution of Evolution," *New Scientist* 2744: 34–47, 26 January 2008. Available 31July2010 at www.newscientist.com/article/mg20527441.500-horizontal-and-vertical-the-evolution-of-evolution.html.

[24] Jean Carlson and John Doyle, "Highly Optimized Tolerance: Robustness and Design in Complex Systems," *Physical Review Letters* 84 (11): 2529–2532, 13 March 2000.
[25] Rick Dove, Patterns of Self-Organizing Agile Security for Resilient Network Situational Awareness and Sense-Making, submitted paper, draft June 2011. www.parshift.com/Files/PsiDocs/PatternsForResilientNetworks.pdf
[26] Daniel T. Blumstein, "Flush Early and Avoid the Rush: A General Rule of Antipredator Behavior?" Behavioral Ecology, 21: 440-442, 26 March 2010.
[27] Juan Camilo Bohorquez, Sean Gourley, Alexander R. Dixon, Michael Spagat and Neil F. Johnson, "Common Ecology Quantifies Human Insurgency," *Nature*, 462(7275), 17 December, 2009, pp 911-914.
[28] Dan Lockton and David Harrison, "Design for Sustainable Behaviour: Investigating Design Methods for Influencing User Behaviour," Sustainable Innovation 09: Towards a Low Carbon Innovation Revolution, 14th International Conference, Farnham Castle, UK, 26-27 October 2009.
[29] Barth F. Smets and Tamar Barkay, "Horizontal Gene Transfer: Perspectives at a Crossroads of Scientific Disciplines," Nature Reviews, Microbiology, Vol. 3, September 2005, pp 675-678.
[30] Jean Carlson and John Doyle, "Complexity and Robustness," PNAS 99: 2538–2545, 19 February, 2002.
[31] Paul Hartzog, "How Different is Your Bow Tie?" Blog at P2P Foundation, 21 June 2010. Available 31July2010 at http://blog.p2pfoundation.net/how-different-is-your-bow-tie/2010/06/21.
[32] Marie Csete and John Doyle, "Bow Ties, Metabolism and Disease," TRENDS in Biotechnology 22(9), September 2004. Available 31July2010 at www.cds.caltech.edu/~doyle/CmplxNets/Trends.pdf.
[33] Aihong Li, et al., "Utilization of Ig Heavy Chain Variable, Diversity, and Joining Gene Segments in Children with B-lineage Acute Lymphoblastic Leukemia: Implications for the Mechanisms of VDJ Recombination and for Pathogenesis," *Blood, 103(12) 4602-4609, 15 June* 2004.
[34] Marie Csete and John Doyle, "Reverse Engineering of Biological Complexity," Science, 295(5560): 1664-1669, 1 March 2002.
[35] Rick Dove and Garry Turkington, "On How Agile Systems Gracefully Migrate Across Next-Generation Life Cycle Boundaries," *Global Journal of Flexible Systems Management*, 10(1), pp 17-26, 2009. www.parshift.com/Files/PsiDocs/Pap080614GloGift08-LifeCycleMigration.pdf
[36] Jennifer Bayuk and Georganne John, "Systemic Security," submitted paper, draft May 2010.
[37] Rick Dove, "Pattern Recognition Without Tradeoffs: Low-Cost Scalable Accuracy Independent of Speed." In Proceedings Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH), 3-4 March 2009, Washington, D.C., 255–260. http://www.kennentech.com/Pubs/2009-PatternRecognitionWithoutTradeoffs.pdf.

X. VITA

Rick Dove is co-founder and chair of the INCOSE working group on System Security Engineering, teaches graduate courses in agile and self organizing systems in the School of Systems and Enterprises at Stevens Institute of Technology, is President of Paradigm Shift International, and is a founding partner of Kennen Technologies. He was co-PI on the OSD/Navy funded project that defined agility as the next generation competitive enterprise characteristic; and led the R&D program at the DARPA/NSF funded Agility Forum. Recently he was the PI on two DHS S&T projects at Kennen Technologies to show feasibility of, and commercialize, massively parallel VLSI-based pattern recognition security technology. He wrote *Response Ability—the Language, Structure and Culture of the Agile Enterprise* (Wiley*)*, and has a BSEE from Carnegie Mellon University.