Establishing Security Strategy Using Systems Thinking

John Wirsbinski, john.wirsbinski@incose.org; and John Boardman, john.boardman@incose.org

Thinking About Security

We assert that the way that we currently think about security is inadequate for today's systems and problems. We believe that a systems approach provides a framework for improving our thinking. However, not just any systems approach is likely to succeed. For the purpose of this essay, we define security as a measure that describes the property that emerges from the interactions between a system, its constituent parts, and one or more adversaries. Security is not intrinsic to a system in isolation, but is only meaningfully addressed in the context of the system (and its parts) and a willful (i.e., human) adversary. This explicit incorporation of people as a dynamic element of the system out of which security emerges requires a particular systems approach. Checkland (1999) advances a systems typology that includes natural systems, designed physical systems, designed abstract systems, human activity systems, and transcendental systems. What is important to consider is that while elements of the security problem space may be designed physical systems, natural systems, or designed abstract systems, the system of interest is at its core a human-activity system. This is a fundamental observation because hard systems engineering approaches are at best partially effective in addressing human-activity systems. Over the last few decades, researchers have developed problem-structuring (systems) methods (Rosenhead 1996; Rosenhead and Mingers 2005) including soft systems methodologies (Checkland 1999; Boardman 2006), and systems-thinking approaches (Boardman and Sauser 2008) to deal with these types of systems. Figure 1 illustrates how systems thinking and soft-systems methodology come together to support and strengthen hard systems engineering. It is our position that, in this context, systems thinking can be used to develop the underlying security strategy that supports both ordered and complex systems engineering (Sheard 2007).

You might be tempted at this point to ask, "How does all of this apply to me? I'm a systems engineer or systems architect. If I need security, I will work with my stakeholders to determine the requirements and then hand them off to my security specialist." We believe that security emerges, in large part, from the system architecture. Therefore, security is the systems engineer's responsibility. The systems engineer also has the greatest insight into how the parts come together and how they might be woven into an architecture that is secure — if the definition of *secure* can be clearly articulated. This idea is echoed by INCOSE's Systems Security Engineering Working Group, which declares in its "Manifesto" (Dove and Wirsbinski 2008) that systems engineers are responsible for ensuring the security of their systems against willful human behavior. The "Manifesto" also articulates the need for new principles and practices for designing security into our systems. We assert that engineers can use systems thinking methods to develop the strategic aspects of these new principles and practices.

In this essay, we write using (1) abstraction and (2) system concepts to encourage you to think systemically about security. We ask you to temporarily suspend thinking about the specifics of your system and allow these abstractions to help you discover new ways to think about your system and associated security considerations. To accomplish this goal, we utilize the Boardman "Conceptagon" (Boardman and Sauser 2008) (shown in figure 2) and four security perspectives.

This brief essay is not the venue to fully articulate the methodology being espoused. In practice, one would examine a system of interest through each lens (dimension) of the Conceptagon. The four security perspectives described in this essay would filter each of these views. Utilizing these views (analogous to architectural views) and filters provides a systematic approach to mining the

» continues on next page

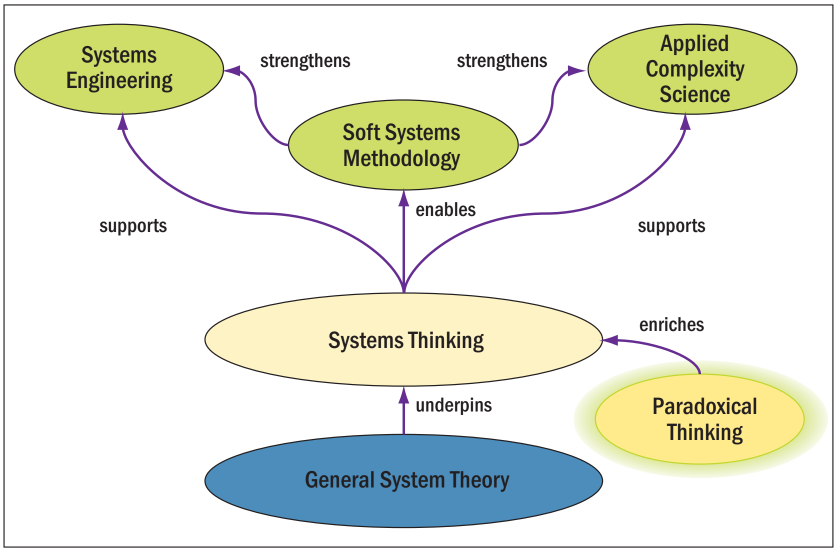


Figure 1. Relationships between systems thinking, soft systems, and systems engineering (Boardman 2008)

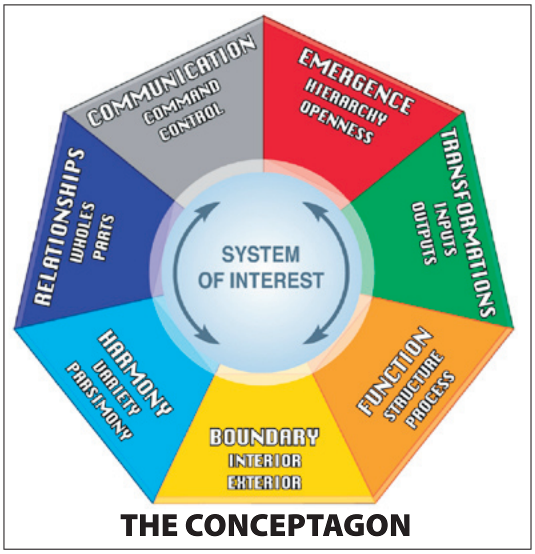


Figure 2. The Boardman Conceptagon (Boardman and Sauser 2008)

security needs and structuring them into a strategy for realizing adequately secure architectures and systems. The remainder of this essay demonstrates some of the types of insight that may be obtained with this approach, using selective examples and systems concepts from the Conceptagon.

Four Security Perspectives

We present four security perspectives: asset, protection objective, adversary, and heuristics/metrics. While time may prove that these four are insufficient to fully characterize system security, we believe that they do comprise a necessary, minimum set.

Asset

The asset perspective is often articulated by the question, What are your assets? This activity draws one or more *boundaries* around the *parts* of greatest value to the system. From this perspective, the function of security is to secure the critical assets. This *function* is often achieved using a fortress-like architecture (e.g., fences, firewalls): one builds a wall around the important assets to keep the adversaries at bay. The selected articulation (deliberate or not) of critical parts and security function often drive aspects of the system architecture. Alternative descriptions of parts and function result in different approaches to security such as multiple, redundant assets strategically separated (survivable) or rapid reconstitution of asset (resilience). The common factor between these approaches is that the security strategy is built around the idea of protecting identifiable system parts.

Protection Objective

The protection objective perspective is closely linked to the asset perspective. Questions often used to explore this perspective are, What are the undesirable events? and, What are the associated consequences? Looking through the lens of the protection objective naturally suggests starting with the idea of *function*. The combination of asset, undesirable event, and consequence often leads engineers to think of security as the *function,* and thus we design our structures and processes to achieve this function (e.g., guns, guards, gates, firewalls, passwords). If the system of interest is a security system, this may be appropriate; however, for most systems engineers this mindset is inappropriate. The protection objective helps to explore this distinction. Security is about protecting something of value. In a designed physical system, the system's function (its purpose) is often the asset that is to be protected. Security is clearly not the primary function and may not be a system function at all. It may only be a system constraint. A more productive mindset for the system engineer is one in which security is used as a measure applied to system structure and process. The concept of security can be used to explore the state (secure or insecure) of the structure, process, communication, transformation, and function of a system of interest sequentially and in combination. Thinking in this manner opens up options for system architecture that do not necessarily involve security measures; these architectural variations would likely be missed if the function were thought to be security.

This perspective, combined with the concept of *exterior,* facilitates exploration of additional protection requirements. When only thinking about what needs to be protected, it is often assumed that everything to protect is inside the system of interest—that is, the assets of value are parts of the system. However, it may be necessary to protect things exterior to the system. It may be necessary to assure that an adversary cannot utilize the system of interest to cause harm to these things exterior to the system. Alternatively, our system may accept inputs that an adversary transforms into vulnerabilities; the vulnerability may reside in the exterior networks that link our system to other systems (Robb 2007). These variations require different tools and architectural approaches from problems that involve protecting parts of our system. They are also unlikely to be revealed by stakeholder elicitation based upon identifying assets interior to the system.

Adversary

The adversary perspective raises interesting architectural implications when used in conjunction with *hierarchy*. Attacks are not random events; adversaries evolve their attacks in response to our systems. As *variety* in adversary threats increases, what type of hierarchy should the system architecture support? A top-down hierarchy strives to achieve security through limiting the variety within the system. Limiting system variety allows the systems engineer to simplify the system architecture and increase control of system structure and process, thereby increasing the likelihood of "designing out" vulnerabilities. Commonality also reduces maintenance issues and simplifies system operation, both of which may be sources of vulnerability. However, this commonality also means that when a vulnerability is discovered, it may be common to all parts of the system. As a result, a single vulnerability may rapidly propagate through our system and many others resulting in large systemic failures. Anyone remember the Love Bug virus? Alternatively, a system architecture with greater openness, flat interactions (Dove 2001), and distributed command, control, and communication may provide greater variety in responses to adversary attacks. Greater variety may also allow one to draw boundaries around subsystems and allow for the compartmentalization of vulnerabilities or even attacks. This approach may make systems more resilient or survivable when attacked. For self-organizing systems, the parts may evolve tailored, emergent responses to attacks. Critically thinking about a system's design hierarchy can reveal implications about how a system is protected. The appropriateness of choices, from a security perspective, can be partially revealed by examining the system hierarchy through the adversary filter.

Heuristics/Metrics

Heuristics and metrics are important tools that help us realize system objectives. The systems concepts in the Conceptagon can help avoid security blunders by helping us think through our heuristic/metric selection. One example of how this is accomplished is to examine common security heuristics such as "protection-in-depth" (i.e., layered protection). This heuristic can be applied to the system architecture as a measure of the system's security. However, a series of layers does not provide significantly more protection than a single layer if all layers share a common vulnerability. Looking at the

» continues opposite

Wirsbinski and Boardman *continued*

heuristics and metrics using the *variety* concept might reveal this problem. In the case of protection-in-depth, it may be possible to show that increasing variety within a few layers may provide equal or greater security to more layers with less variety. These types of trade-offs clearly reside under the purview of the systems engineer; the Conceptagon and the security perspectives provide a systematic mechanism to explore the trade-space.

Closing Thoughts

Helen Keller wrote, "Security is mostly a superstition. It does not exist in nature, nor do the children of men as a whole experience it. Avoiding danger is no safer in the long run than outright exposure. Life is either a daring adventure, or nothing" (Keller 1957). Perfect security is a myth, and though security should not override all other considerations, it should be part of the systems engineering trade-space. Systems thinking and security perspectives provide a toolset that enables systems engineers and architects to achieve systems objectives in a manner that is appropriately secure.

References

Boardman, J. 2006. *SDOE 775 Systems thinking: unit 5; soft systems*. Hoboken, NJ: Stevens Institute of Technology.

———. 2008. Systems thinking. Paper presented at Sandia National Laboratories, Albuquerque, NM.

Boardman, J., and B. Sauser. 2008. *Systems thinking: Coping with 21st century problems.* Boca Raton, FL: CRC Press.

Checkland, P. 1999. *Systems thinking, systems practice.* New York: Wiley.

Dove, R. 2001. *Response ability: The language, structure, and culture of the agile enterprise.* New York: Wiley.

Dove, R., and J. W. Wirsbinski. 2008. The manifesto of the working group on systems security engineering: A declaration of responsibility. *INSIGHT* 11 (2): 47–49.

Keller, Helen. 1957. *The Open Door.* Garden City, NY: Doubleday.

Robb, J. .2007. *Brave new war: The next stage of terrorism and the end of globalization.* New York: Wiley.

Rosenhead, J. 1996. What's the problem? An introduction to problem structuring methods. *Interfaces* 26 (6): 117–131.

Rosenhead, J., and J. Mingers. 2005. *Rational analysis for a problematic world revisited.* New York: Wiley.

Sheard, S. A. 2007. Principles of complex systems for systems of systems engineering. Paper presented at the 17th Annual International Symposium of INCOSE, San Diego, CA.

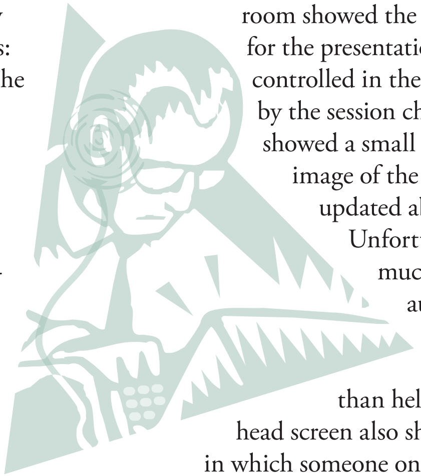
Fellows' Insight

Using Technology to Access a World of Speakers for Chapter Meetings

Joseph Kasser, joseph.kasser@incose.org

It seems that many INCOSE members who find themselves faced with the task of organising programs for their chapters end up working in isolation, and eventually giving up in frustration. However, my research and experience have shown that technology can provide a solution to the problem and provide world-class speakers at little cost and not much more effort. This is because most presentations, be they at chapter meetings or conferences, follow the same format or process:

1. The host introduces the speaker.
2. The speaker gives the presentation.
3. A discussion takes place between the speaker and the audience.
4. The host and audience thank the speaker.



As long as no one interrupts the speaker during the talk, we can ask, does the speaker actually need to be in the room? I have researched this problem since 1998 and the answer seems to be no, not if the host group uses appropriate technology. The possibilities of remote presentations open the door for chapters to host speakers from anywhere in the world.

What, then, is the appropriate technology? As a systems engineer, I must say that this is a very difficult question to answer in general, since we have not yet defined the requirements for the situation; but I have documented several approaches that can help chapters host speakers remotely, including those described below. Each of them has worked for me, allowing the provision of an interesting talk to an audience geographically separated from the speaker.

In May 1999, I watched Doug Vogel of City University of Hong Kong, make a live presentation from Hong Kong to an audience in Virginia Beach, Virginia, at the Fourth Annual Intelligence Community Desktop Collaboration Conference and Exposition; he used PowerPoint for the slides and Microsoft NetMeeting software. One screen in the conference room showed the PowerPoint slides for the presentation, which were controlled in the conference room by the session chair. A second screen showed a small "talking head" image of the presenter, which was updated about once a second. Unfortunately there was not much movement for the audience to see, so the image seemed to be more distracting than helpful. The talking head screen also showed a text window in which someone on the receiving end in Virginia could type questions back to the presenter, but only the session chair could type into the communications link. While there were several noticeable interference hits on the presenter's voice during the presentation, the link was reasonably clear, and there was little difference between that technology-enhanced presentation and a conventional presentation.

During the INCOSE International Symposium in Brighton, U.K., in 1999, I made a presentation to a panel forum from my desk at the University of Maryland University College (UMUC), located about twenty miles from the White House in Washington, DC. I recorded my presentation before the symposium, put it on a CD-ROM, and sent it to the session chair before the

» *continues on next page*