Managing the Risks of Cyber-Physical Systems

C. Warren Axelrod, Ph.D., *IEEE Member*
Decilog Inc.
Melville, New York 11747

***Abstract*—We are increasingly seeing the merging of information systems and industrial control systems into so-called cyber-physical systems; the smart grid being a prime example. This trend leads to major risk issues because the viewpoint of those designing and developing security-critical information (or computational or business) systems differs markedly from how those creating safety-critical control systems consider hazards and resulting risk. Essentially, information security has to do with protecting information assets, such as intellectual property and sensitive personal information, from falling into the hands of those bent on fraud and other nefarious activities. On the other hand, the focus of those responsible for the safety of software-intensive systems are intent on ensuring that a system malfunction or failure will not harm humans or the environment.**

By combining security-critical information systems and safety-critical control systems, we have been creating a risk environment for these computer systems that is greater than the sum of the risk of the parts. For example, industrial control systems traditionally have been isolated from public networks and therefore not subject to cyber attacks over the Internet. As a consequence, such systems as these were never designed to withstand such remote attacks and are generally more vulnerable than information systems. On the other hand, those responsible for security-critical software systems would typically not consider physical harm resulting from successful attacks and believed that the worst that might happen would be financial losses. In the new cyber-physical systems world, designers and developers have to be concerned about the possibility of their systems being used as a conduit to controlling systems that have national security and critical infrastructure ramifications.

In this paper we look at the totality of risks across a broad range of cyber-physical systems in the public and private sectors and point to areas that must be subjected to much greater scrutiny in order to mitigate increased risks. Since the risk is greater than the sum of the parts, so the mitigating activities must be that much greater to the extent that any security/safety approach needs to account for the risks of not only the individual components but also of the interactions among the components. This might well facilitate the justification of much greater expenditures and effort on securing the overall system since the consequences of successful breaches is that much greater.

We present a model that helps to determine the factors that lead to levels of combined risk and will propose appropriate methods to suitably contain and minimize such risk.

Keywords-risk; hazards; cyber-physical systems; vulnerabilities; threats; exploits

I. INTRODUCTION

As described in Axelrod [1], there are major cultural and orientation differences between software engineers responsible for safety-critical software-intensive systems and those responsible for security-critical systems. This is in large part due to the requirement for security-critical systems to protect sensitive information (such as nonpublic personal information, and health-related data), intellectual property, and the like, versus the need to ensure that safety-critical systems (such as avionics software and software running on industrial control systems) do not harm people or the environment. Because these orientations are so different, and may have little overlap, the threats to these systems, their vulnerabilities and the consequences of breaches, malfunctions and failure are also very different. This is illustrated at a high level in Figure 1.

II. DIFFERENTIATING BETWEEN SOS AND CPS

A. System of Systems

Definitions of a "system" and "system of systems" per the INCOSE (International Council of Systems Engineering) Systems Engineering Handbook [2] are as follows:

> "A system is a combination of interacting elements organized to achieve one or more stated purposes."

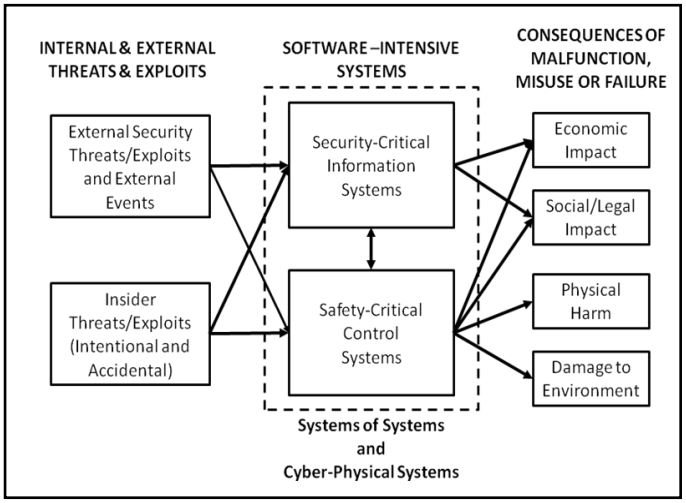

Figure 1. Consequences of malfunction, misuse or failure
(From C.W. Axelrod, ***Engineering Safe and Secure Software Systems***, © 2013 Artech House)

© 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

> "System of systems applies to a system-of-interest whose system elements are themselves systems; typically these entail large scale inter-disciplinary problems with multiple, heterogeneous, distributed systems."

Another definition of a system of systems (SoS), from the U.S. Department of Defense [3], combines the definitions of a system and an SoS as follows:

> "*An SoS is defined as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities* ... Both individual systems and SoS conform to the accepted definition of a system in that each consists of parts, relationships, and a whole that is greater than the sum of the parts; however, although an SoS is a system, not all systems are SoS."

Reference [3] distinguishes between SoS and FoS (Family of Systems) as follows:

> "... FoS are fundamentally different from SoS because ... *a family of systems lacks the synergy of a system of systems. The family of systems does not acquire qualitatively new properties as a result of the grouping.* In fact, the member systems may not be connected into a whole ..."

B. Cyber-Physical System

The National Science Foundation [4] defines and describes cyber-physical systems as follows:

> "The term cyber-physical systems refers to the tight conjoining of and coordination between computational and physical resources. We envision that the cyber-physical systems of tomorrow will far exceed those of today in terms of adaptability, autonomy, efficiency, functionality, reliability, safety, and usability. Research advances in cyber-physical systems promise to transform our world with systems that respond more quickly (e.g., autonomous collision avoidance), are more precise (e.g., robotic surgery and nano-tolerance manufacturing), work in dangerous or inaccessible environments (e.g., autonomous systems for search and rescue, firefighting, and exploration), provide large-scale, distributed coordination (e.g., automated traffic control), are highly efficient (e.g., zero-net energy buildings), augment human capabilities, and enhance societal wellbeing (e.g., assistive technologies and ubiquitous healthcare monitoring and delivery)."

While the definition of CPS does not specifically include the term "system of systems," it is highly likely that most cyber-physical systems will be in the systems-of-systems category, in that the whole is greater than the sum of the parts. Conversely, all systems of systems do not necessarily contain physical systems and therefore may not be considered to be cyber-physical systems.

It is interesting to note that, while many CPS characteristics are listed (e.g., adaptability, reliability, safety), "security" is notably omitted. As will be discussed in this paper, the security aspects of aggregated systems become increasingly important as control systems are interfaced with cyber systems. We shall refine the model shown in Figure 1 and investigate the risks related to each of the boxes in detail.

C. A Further Distinction

While the NSF definition of CPS in [3] is very detailed, it is held by this writer that the mere distinction between computational and physical systems does not adequately describe the types of system in which we are interested. This is because cyber systems (informational or computational or business systems) have physical components, while industrial control systems (such as SCADA systems – Supervisory Control and Data Acquisition systems) include some data processing functionality to handle the administrative and control functions. These latter systems are often termed "embedded systems" since they incorporate specialized functions in the software and hardware specific to the physical systems being managed by them. Thus an elevator control system may be written in machine language running on specially-developed computer chips and/or specially-programmed firmware. This is illustrated in Figure 2, where the embedded system is included within the context of the physical system.

Consequently, we will distinguish between the cyber and physical components of CPSs by taking the position that the cyber components are purely for information processing purposes and may be accessed over a public network, whereas the primary function of physical systems is to control physical devices. This is shown in Table I.

Put another way, both cyber and physical systems have information processing and physical equipment components. For cyber systems, the applications software that processes the information is the primary function of the system, whereas the hardware is to support the software. For industrial control systems, such as SCADA systems, the information processing system supports the equipment being controlled. To make the description more complicated, but more accurate, it should be noted that the monitoring and administrative software of control systems will also reside on computer hardware.

TABLE I. COMPUTATIONAL AND PHYSICAL COMPONENTS FOR CYBER-PHYSICAL SYSTEMS RESOURCES

	Resources	
Components	***Computational systems***	***Physical systems***
Cyber systems	Data processing	Supporting hardware
Physical systems	Supervisory control systems	Mechanical devices

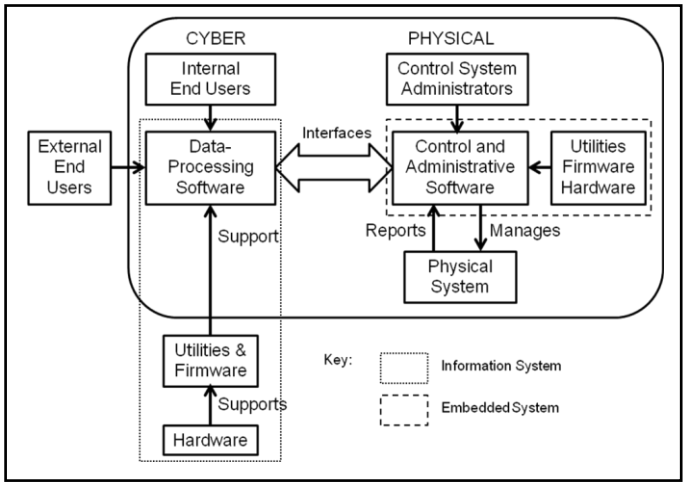

Figure 2. Cyber and physical components of cyber-physical systems

Usually cyber systems are accessible by numbers of internal and external end users, whereas access to SCADA software is limited to those whose function it is to operate the physical system in question. This is illustrated in Figure 2.

This distinction is perhaps best illustrated by the example of the Stuxnet worm which effectively destroyed centrifuges processing uranium at an Iranian facility.[1] It was reported that the SCADA system, which monitored and controlled the uranium enrichment process, was attacked via an infected USB drive which was carried into the facility by an authorized individual. Not only did the malware cause the centrifuges to self-destruct, but the supervisory control system, which monitored the performance of the centrifuges was compromised so that it displayed on the operators' consoles that the centrifuges were operating normally when in fact they were not.

III. Risk Mitigation

In this section we look at how to measure risk, which factors contribute to risk in the cyber-physical model (including those risks that are unique to the CPS environment), and how such risks can be avoided or mitigated if they cannot be prevented.

A. Approach to Risk Evaluation

Many different methods for calculating risk appear in the literature, as discussed in Axelrod [1]. They range from the simplistic to the complex. This writer prefers using "expected loss" as the measure of risk. It is the product of the magnitude of loss that would be experienced were an event to happen and the probability of occurrence. The magnitude of a loss will usually include direct costs, such as compensating credit-card holders for fraudulent activity in their accounts, but should also take into account indirect and intangible costs, such as reputation loss. The probability of a loss-producing event is derived or estimated from knowledge of threats, exploits and vulnerabilities.

As an example, if a data breach would cost $1 million and the probability of it occurring over a given period, say one year, is 5 percent, then the expected loss is $1 million multiplied by 0.05, which equals $50,000. This means that an organization might be willing to pay up to $50,000 to prevent such an event from occurring.

While the above method might appear to be easy to use, difficulties arise in obtaining accurate estimates of the magnitude and probability of loss for a particular event. Much of the difficulty on the magnitude side is that, in many cases, losses are suffered by a number of different users (customers, operators, manufacturers, etc.), each with their own perspectives. For example, a company may not disclose the full extent of the damage (unless they are required to by law) in order to minimize claims against it. Also, certain forms of damage, such as loss of reputation and physical and mental harm to individuals, can be extremely difficult to measure.

On the probability side, it is very difficult to predict infrequent but high impact events. These types of event, which are described as "black swans" by Taleb [5], occur with some regularity, but tend to be different in every case in terms of human life, economic resources, magnitude, location and impact.

However, despite the problems of formulation, estimation and determination, such an approach tends to be superior to many other approaches and certainly is better than doing nothing.

B. Risk Model Factors

In order to arrive at reasonable estimates of the probability and magnitude of adverse events, it is helpful to have in mind a model as to what actually happens when an attack is launched or something bad just happens to a security-critical (cyber) system. Such a model is shown in Figure 3.

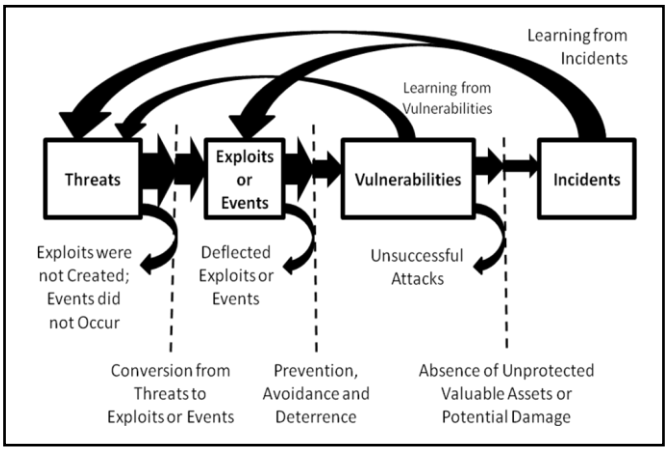

Figure 3. Threats, exploits, events, vulnerabilities and incidents relating to security-critical cyber systems
(From C.W. Axelrod, ***Engineering Safe and Secure Software Systems***, © 2013 Artech House)

[1] See http://en.wikipedia.org/wiki/Stuxnet for a description of Stuxnet, its implementation and consequences.

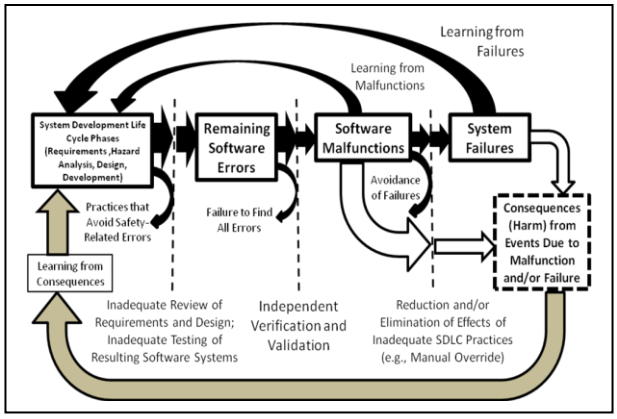

Figure 4. Reduction and/or elimination of errors that might result in malfunction and/or failure

A comparable diagram for safety-critical systems is shown in Figure 4. However, the differences in the diagrams of Figures 3 and 4, respectively, are informative. Figure 3 is oriented towards effectively creating a protective shield around security-critical software systems so that attackers are thwarted, whereas Figure 4 shows that the approach for safety-critical systems is more introspective with effort put into the design and implementation of the systems themselves.

Figure 4 is meant to show that design and programming errors can cause a software system to malfunction and possibly fail. Consequential harm can be done to humans and the environment by a malfunction that does not lead to system failure and by failures themselves. The impact of such harmful events also determine the degree to which lessons learned feed back from the malfunctions and failures themselves and the extent to which actual incidents occur. That is to say, an engine failure can occur when a vehicle is stationary or travelling at speed. In the former situation, little damage may be incurred, whereas in the latter, there could be extensive loss of life. When the consequences are severe, the reaction and response will generally be much greater than were the consequences to be minimal. This really depends on whether the analysts extrapolate to the various contexts in which a malfunction or failure might occur. Sometimes a "near miss" is taken as an indication that damaging consequences will be minimal or nonexistent in most cases; whereas the "near miss" might have been a one-in-a-million case and the greatest likelihood is that extensive damage will usually result from such a malfunction or failure.

Implicit in Figure 4 are the hazards that might result from malfunctions and failures; their impact and the likelihood that those hazards could occur.

For cyber-physical systems, the attacks on the cyber systems and the potential harm from malfunctions and failures of the physical control systems are combined. The impact of such attacks and malfunctions/failures is exacerbated due to lack of consideration of harmful failures by those designing cyber systems and not considering the possibility of a cyber attack flowing though to control systems by those designing control systems.

IV. Enumeration of Threats to Information Systems

An extensive list of the threats to which information systems are subject is given in BITS [6], page 6. The threats are broken down into the following categories:

- Human actors (outsiders, insiders)
  - Network access – external
    - Motive – deliberate (A), accidental (B)
  - Network access – internal
    - Motive – deliberate (C), accidental (D)
  - Physical access – external
    - Motive – deliberate (E), accidental (F)
  - Physical access – internal
    - Motive – deliberate (G), accidental (H)
- Non-human actors
  - System access – external (I), internal (J)
  - Natural access – external (K), internal (L)

The table in BITS [6], even though it is a sample inventory of threats, has about 100 threats listed. Here is a small sample by category, as tagged above:

A. DDoS attacks, website defacements
B. Unintentional DDoS attacks
C. Network/applications time bombs
D. Human error
E. Terrorist attacks
F. Radiation contamination
G. Embezzlement, sabotage
H. Leaving doors unlocked
I. Power failure
J. System software failure
K. Hurricanes, tornados
L. Fire, floods

It should be noted that the originators of the above threats are internal and external actors and not the computer systems and networks. When computer systems and networks are the origins of such threats, we are dealing with safety-critical control systems and their potential impact on humans and the environment.

V. Enumeration of Hazards from Control Systems

A number of the hazards listed below are the same as those potentially experienced by information systems with the difference that, with control systems and computer and network hardware, the hazardous situations are caused by malfunctioning or failure of the physical systems. At the same time, control systems can be subject to the same threats as

information systems. As an example, a fire in a nuclear power plant could cause failure of control systems, but a malfunction or failure of a control system could result in a fire. Indeed, it can happen that an event, such as the tsunami in the Fukushima catastrophe on March 11, 2011, which knocks out control systems, results in failures leading to other hazards, such as explosions and fires. Many of the threats enumerated in the BITS report [6], also apply to physical systems. However, in this section, we identify hazard risks ***resulting from*** failed or malfunctioning control systems, whether caused intentionally or accidentally. Here is a brief list of potential hazard risks:

- Explosions, fires, floods, etc. resulting from failures of power plants, refineries, faulty computer controls, batteries[2]
- Mechanical failures of generators, centrifuges, etc.
- Floods, e.g., from failed systems controlling dams, water treatment plants, waste treatment plants
- Chemical, biochemical spills and releases
- Potential crashes and other accidents of vehicles such as aircraft, ships, trains, automobiles, etc. resulting from system malfunctioning or failures
- Malfunction or failure of weapon systems, such as artillery, missiles, drones (UAVs – Unmanned Aerial Vehicles) and resulting unintended harm.

VI. Risks by System Component

The above threats and hazards may be used to estimate the risks to the software systems that they affect or are affected by. The BITS report [6] provides a useful methodology for assessing and quantifying the risks to which information systems are subjected. Rushby [7] provides a helpful framework for evaluating hazard risks incurred from deploying and operating safety-critical systems.

It may be theoretically feasible to estimate the risks to the information systems and from the control systems and combine them to determine the total exposure to and from of a cyber-physical system. However, such an approach would greatly underestimate the overall risks. As discussed above, there is a need to include those risks to the physical systems that derive from being connected to cyber systems, and vice versa. Once that has been done—and it is no easy task—then someone or some group needs to assume responsibility for mitigating these supplementary risks. Persuading someone to do this task is also difficult to accomplish in real-world situations.

[2] As of early February 2013, the root cause of the Boeing Dreamliner 787's batteries catching fire had not been identified, see A. Pasztor and J. Ostrower, "787 Fires Still a Mystery: Investigators Struggle to Discern Root Cause of Dreamliner's Burning Batteries," The Wall Street Journal, February 5, 2013, pages B1 and B2.

VII. Risks and Liabilities from Combining Systems

When information and control systems are combined into cyber-physical systems, they are usually subjected to threats of attacks on their information systems along with the potential for harm to humans and environment from physical systems managed by malfunctioning or failed control systems.[3]

However, it is particularly important to take account of the risks of combined information and control systems since, not only are the risks of a cyber-physical system the sum of the risks of each component system, but there is also a synergistic risk effect, which is particularly reflected in the liability that is assumed for the information systems and control systems and how that liability is extended for combined systems.

As an example, let us take the case of driverless cars, which are a rapidly growing example of combining information and control systems, and the impact on liability on a company such as Google, which is at the forefront of research in this area. Google is a software company. Software use agreements explicitly state, in capitalized print, that the vendor assumes no liability for the correct working of the software and is not responsible for consequential damages. That is to say, if software (or a service based on software) were to malfunction or fail, all most software vendors will agree to is to fix the problem (usually with a patch) or refund the cost of the software. Contrast this with the liabilities assumed by automobile manufacturers. If a safety problem arises, the manufacturer must actively attempt to recall all affected vehicles and fix them. There have already been issues with the software systems controlling vehicles; a prior example being Toyota's issues with uncontrolled acceleration, although the role of software errors was never confirmed. However, it should be noted that the systems under scrutiny in the Toyota case is in the "control and administrative software" category shown on the right-hand side of Figure 2, and not the "data processing software" on the left of the diagram.

Note, however, that by interfacing its navigation system, which is a data processing system, with control systems, such as those that control the speed of the vehicle, braking, turning, etc., all of a sudden, there arises a major liability issue since a failure could lead to a fatal accident. This issue is beginning to be debated as in Strumpf [8], and could well hamper the adoption of autonomous vehicles on the current road system.

VIII. Minimizing the Total Risk

It is clear from the above that the creation of cyber-physical systems increases the risks from those systems to a greater degree than the sum of the risks of the individual component systems, so that, even if we were to assume that mitigation strategies and tools were in place to minimize the

[3] Information systems can malfunction or fail from events affecting the hardware platforms on which they run and physical failures can occur independently of their control systems. However, these instances are not within the scope of this article.

individual system's risks (which they are not), then we still have to address the increased risk from combined systems.

A. *Improving the Quality of Component Systems*

An initial means of addressing the overall level of risk for a cyber-physical system is to improve the quality of the design and testing of the component systems. This can be justified by the increase liability alone. Now an error in a data-processing system might lead to harm to humans and/or the environment, as could a compromise of these systems. Certainly, a first step is to build security into the development lifecycle and introduce much more rigorous testing of the data-processing system, as described in Axelrod [10]. Such design and testing aspects need to comprehend an environment in which the outputs of the data-processing system influence control systems. A parallel effort needs to be introduced on the control systems side. Design and testing must include awareness that data will be introduced from external systems (as well as from and to the administrative and control systems operating the physical systems), and increase the security requirements as well as the verification and validation efforts correspondingly.

B. *The Importance of Integration Testing*

However, as stated above, even when the component systems have been hardened with respect to security and safety, there are new conditions introduced through the combination of systems. This calls for an additional process dedicated to the requirements, design and testing of the combined systems. That is to say, the ways in which the combined cyber-physical systems can be attacked, malfunction or fail are greatly increased and are unlikely to be recognized when examining the individual system. Collaboration among the various security and safety software engineers is essential for this, as described in Axelrod [1].

C. *Fail-Safe and Fail-Secure Requirements*

Particular focus needs to be applied to the behavior of the individual and combined systems when they malfunction and fail. This is becoming increasingly important as manual control by experienced human operators is being reduced. Further, more systems are becoming so complex and interconnected that it is almost impossible for any individual to comprehend all aspects of modern systems of systems. As a consequence, it is not reasonable to depend on human reactions to unusual and unexpected events affecting or being affected by these systems. This calls for the need to ensure that security and safety requirements are clearly annunciated at the early stages of the development life cycle, and that these requirements are incorporated into the systems and fully implemented and tested for a variety of situations.

Sometimes there is a conflict between failing safe and failing secure. For example, if a building entry system fails, security practitioners might prefer that it fails closed so that no unauthorized person can enter the building, whereas safety engineers would want the system to fail open so that persons inside can get out, as one would want if there was a fire.

D. *Integrating Human Beings into Cyber-Physical Systems*

The human component in cyber-physical systems has to be understood and accounted for, as described in Schirner [9].[4] For example, the role of humans in "driverless" vehicles has to be assessed and the systems have to allow for appropriate human overrides and other involvement. Unfortunately, in much of the literature, humans' contribution in the loop is treated minimally or not at all.

IX. Summary and Conclusions

As systems are developed and combined to form cyber-physical systems, the risks of the whole system are greater than the sum of the risks of the component systems. In order to assess and mitigate these overall risks, it is necessary to understand the threats to data-processing systems and the harm that could emanate from software that controls physical systems. This article has attempted to enumerate the entirety of the risks and suggests how they might be reduced or, preferably, eliminated in safety-critical systems. The suggested mitigation strategies include substantial improvements in how current systems are designed, developed and tested, and a process for ensuring that the combined systems meet equally strict security and safety requirements.

References

[1] C.W. Axelrod, Engineering Safe and Secure Software Systems, Norwood, MA: Artech House, 2012.

[2] International Council of Systems Engineering, INCOSE Systems Engineering Handbook—A Guide for System Life Cycle Processes and Activities, Version 3.2, 2010. Available for purchase at http://www.incose.org/ProductsPubs/products/sehandbook.aspx

[3] Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. Systems Engineering Guide for Systems of Systems, Version 1.0. Washington, DC: ODUSD(A&T)SSE, 2008. Available at http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf

[4] National Science Foundation (NSF), Cyber-Physical System (CPS), Program Solicitation NSF 10-515, 2010. Available at http://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm

[5] N.N. Taleb, The Black Swan: The Impact of the Highly Improbable, 2nd ed., Random House, 2010.

[6] BITS, Kalculator: BITS Key Risk Measurement Tool for Information Security Operational Risks, 2004. The report is available at http://www.bits.org/publications/doc/BITSKalcManage0704.pdf

[7] J. Rushby, Critical System Properties: Survey and Taxonomy, SRI International, Technical Report CSL-93-01, 1993. The report is available at http://www.csl.sri.com/users/rushby/papers/csl-93-1.pdf

[8] D. Strumpf, "Liability Issues Create Potholes On the Road to Driverless Cars," The Wall Street Journal, January 28, 2013, pp. B1 and B7.

[9] G. Schirner, et al, "The Future of Human-in-the-Loop Cyber-Physical Systems," IEEE Computer Journal, Vol. 46, No. 1, January 2013, pp. 36-45.

[10] A.K. Chaudary, "Is the Business Network Connected to SCADA? Need for Auditing SCADA Networks," JOnline, ISACA Journal, Vol. 6, 2012. This article is available (ISACA membership required) at https://www.isaca.org/ecommerce/Pages/Login.aspx?ReturnUrl=%2fJournal%2fPast-Issues%2f2012%2fVolume-6%2fPages%2fJOnline-Is-the-Business-Network-Connected-to-SCADA.aspx

[4] Schirner [9] has a different definition of a cyber-physical system (CPS) from that in Figure 2 of this article. He suggests that a CPS consists of an embedded system and physical sensors and actuators. In this article, we differentiate between information systems and embedded systems and require that a CPS includes information (or computation) systems and not just embedded systems. This latter view is also expressed in Chaudary [10].