New Competencies for Control Engineers to Meet the Market Demands in Control Systems

Mariana Hentea, Harpal Dhillon
Excelsior College, Albany, USA
mhentea@excelsior.edu

The increased connectivity to Internet and mobile device technology has a major impact on control systems software design and system architecture. To take advantage of new market opportunities and emerging technologies, these systems/products must evolve very rapidly. Control systems are becoming very complex software applications, and the software engineers responsible for designing and maintaining these control systems must be trained innovatively. This paper discusses the complexity of new generation of control systems and the need for greater compliance to safety, quality of service, and security of systems and data. All these issues impact the education and training of software developers and control engineers. Software developers are required to ensure that software is stable and reliable to avoid hazard conditions. Besides mastering control methods, the control engineers must be able to deal with the challenges created by emerging technologies, and the capabilities of the users of these complex control systems. These challenges demand adoption of new emerging technologies and new competencies and skills. These competencies allow open possibilities for applying a wider range of control principles and design methods as well as productivity control and risk management involving decision making under uncertainty with increased levels of decision support.

Index Terms – Control engineers, Control systems, Knowledge, Skills, Complex systems, Computer Engineering, Systems Engineering.

INTRODUCTION

A system is defined as "an interacting combination of elements to accomplish a defined objective. These include hardware, software, firmware, people, information, techniques, facilities, services, and other support elements," [International Council on Systems Engineering [1]. A control system is a device or set of devices to manage, command, direct or regulate the behaviour of other devices or systems. Examples of control systems include Supervisory Control And Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other smaller control system configurations including skid-mounted Programmable Logic Controllers **(**PLC**).** These are also known under a general term, Industrial Control System (ICS). The control systems are often found in the industrial sectors and critical infrastructures. ICS**s** are typically used in industries such as electrical, water, oil and gas, chemical including experimental and research facilities such as nuclear fusion laboratories. The reliable operation of modern infrastructures depends on these systems. There is a growing concern about the security and safety of the control systems in terms of vulnerabilities, lack of protection, and awareness [2], [3]. Many organizations consider that software security is an important and growing aspect of software quality for control systems.

In the past, control systems were isolated from other Information Technology (IT) systems. Connection to the Internet is new (early 1990s) and debatable among specialists. Many specialists agree that exposing control systems to the Internet is not a good idea. However, without any connection to the Internet these systems are still vulnerable to external or internal attackers that can exploit vulnerabilities in software such as operating systems, custom and vendor software, data storage software, databases, and applications. The increasing use of the Internet, the proliferation of Web technologies, and mobile technology has created a demand for a wide variety of new products. These changes had a major impact on control systems architectures. As organizations rely more on use of control systems, information security concerns have given rise to new software needs. Thus, we need to evaluate and identify means to improve these systems. In this paper, we discuss the complexity of new generation of control systems and the need for greater compliance to safety, quality of service, and security of systems and data. All these issues impact the education and training of software developers and control engineers.

The rest of this paper is organized in sections as follows: we briefly discuss the control systems problems in the next section. Then, we present the new competencies for control engineers in the following section. Next, we discuss a proposal incorporating new approaches for educating and preparing future control engineers and software developers with adequate skills for design and development of secure, user friendly, and effective control systems. We conclude with thoughts about the future.

CONTROL SYSTEMS CHALLENGES

Control systems evolved from static to dynamic systems with access to Internet. For example, the electric power industry is using World Wide Web (WEB) for information exchange, analytical simulations, and actual control of the electric power grid [4]. However, use of the WEB technologies raises several serious issues such as:

- Can the security of communications be assured?
- Can time-critical operations be carried out reliably?
- Can delays be accurately predicted and securely compensated for?

In addition, standardization and use of open market technologies are current requirements in control systems. Modern products are often based on component architectures using commercial off-the-shelf products (COTS) elements as units. This architecture leads to control systems that "are becoming very complex software applications" with the following characteristics [5]:

- Time critical
- Embedded
- Fault Tolerant
- Distributed
- Intelligent
- Large
- Open
- Heterogeneous.

New languages and platforms such as Java, C#, and CORBA are promising increased ease of use, portability, and safety and contribute to making heterogeneous distributed control system platforms possible. Hans-Jurgen Weidemann [6] stressed the complexity of control systems and greater need of compliance for safety, quality of service, and security of systems and data.

Control systems are exposed to the same cyberspace threats like any business system because they share the common vulnerabilities with IT systems. Also, most control systems are not protected with appropriate security safeguards and personnel is lacking the security training and awareness. Threats against control systems are ranked high in the list of government concerns, since terrorists have threatened to attack several systems of critical infrastructure. Cyber interdependencies are a result of the pervasive computerization and automation of infrastructures [7].

Cyber attacks exploit vulnerabilities previously not modeled or unknown to a system. Control systems were initially designed with little attention to security. However, the modern control systems, integrated with corporate networks and the Internet, have become far more vulnerable to unauthorized cyber attacks putting the national infrastructures at risk and easy targets of attacks by terrorists. Besides security concerns, the computer systems including control systems raise the issue of safety causing harm and catastrophic damage when they fail to support applications as intended [8].

A growing number of worms and viruses are spread by exploiting software design, operations, and human interfaces. The software-intensive system design skills for the construction of control systems are often misunderstood. In control industry, two separate groups of engineers are typically involved in the development of any nontrivial controller: control engineers and programmers. These two groups tend to have very different perspectives and working practices, and both lack the global picture needed for the task [7]. These developers are expected to facilitate and open possibilities for applying a wide range of control principles and methods as well productivity control, involving decision making under uncertainty with increased levels of decision support.

Control systems design and information security management must be improved. The improvements for the security of control systems have to be broad - at the systems level, and detailed - at the component level. More efforts should be planned for reducing the vulnerabilities and improving the security operations of these systems. It is necessary to address not only the individual vulnerabilities, but the breadth of risks that can interfere with critical operations. Thus, the academia and organizations need to increase the skills and enhance the knowledge of control systems developers. The following section summarizes aspects of new knowledge and skills for the development of the software for the control systems.

NEW COMPETENCIES FOR CONTROL ENGINEERS

Control engineering topics are usually taught in Electrical and Computer Engineering programs as well as in Mechanical Engineering. A few universities offer graduate programs in control. Organizations hire control engineers for plant automation and development of control systems software. Developing new software for control systems requires new knowledge. However, what else is important today and in the near future? What are very important skills? Specific issues that are required to be addressed by developers of control systems are summarized as follows:

1) Cyber Security

The evaluation of the security of the software is crucial. Software plays an increasingly important role in all types of controllers. Since real-time applications alone may never be capable of addressing all security requirements, incorporating security features into a device can further enhance system security. Individual devices erect their own security perimeters and defend their own critical resources, such as a network link or storage media [9]. Although protecting code in embedded systems is sometimes a business decision, a company must weigh the cost of implementation of protection versus the potential loss of service revenue when the vulnerabilities are discovered [10].

Implementing security controls (countermeasures) is a way of mitigating risks. Therefore, there is a need for increased use of automated auditing and intelligent reporting mechanisms that must support proactive security assessment and threat management [11].

2) *Integration*

Low-level, real-time technology needs to be combined with high-level aspects, such as programming, networking, security, simulation, and control. There is a need for a new framework for the analysis, design, and operation of complex systems that will emerge from the interaction of several disciplines and will include the development of new architectures with learning, adaptation and self-reconfiguration features; new modeling, control, estimation, and monitoring strategies; new computational strategies to deal with complexity; and the integration of control, communications and computing.

3) Intelligent control

Intelligent control is a rapidly evolving, complex and challenging field with great practical importance and potential. Intelligent control systems emerged from artificial intelligence and computer controlled systems as an interdisciplinary field [12]. Intelligent systems can support continuous monitoring and controlling plant activities. Intelligence improves an individual's ability to make better decisions for information security management and decision making with an effect size that is higher than an expert in security by providing mechanisms to enhance the active construction of knowledge about threats, policies, procedures, and risks [13].

Given our reliance on digital technology, control theory has to be broadened. Application of advanced control schemes is becoming the norm. New algorithms such as adaptive controllers , neural network controllers, fuzzy controllers, multivariable controllers, multivariable model predictive controllers, bilinear controller are successfully implemented. The computer has facilitated applications of control that were considered impossible before, with new complex systems such as physically distributed systems, multi-agents, multi-scale behaviors, networked systems, and high levels of integration.

4) Emerging technologies

In order to evaluate the performance of a component, subsystem, or a system that cannot be measured, for example, it is necessary to use analytic or simulation models [14]. Evaluation of computing and telecommunications systems is needed at every stage in the life cycle, including their design, manufacturing, sale/purchase, use, upgrade, tuning, etc. Simulation has been changed since WEB technologies offer a new set of capabilities. On-line statistical analysis tools and on-line intelligent controller for managing a system are needed. Such a controller first needs to determine control strategies, perform the essential on-line simulations that are needed to assess their potential performance, compare their projected responses in order to determine which strategy should be implemented, and then implement the selected strategy. Thus, all of these functions must be addressed concurrently while the system is operating [15].

5) Market demands

New market demands have to be analyzed and implemented. Examples include a fully operational on-line intelligent controller for a real manufacturing cell; network control systems require intelligent adaptation to changing situations, especially tolerance to faults. Self-repair and recovery can be developed for the network control systems.

6) Knowledge and skills

Developers of control systems need new skills and knowledge in many areas. Control engineers require new knowledge and programming skills in addition to engineering skills in order to understand the broad picture and detailed control problems. A common perspective and security solution is essential in design of these systems. New approaches and skills for software development of the distributed control systems architectures are required to ensure that software is stable and reliable to avoid hazard conditions [16]. Control theory has been used to design and analyze feedback loops for computing systems [17]. These efforts have resulted in products that are more robust to disturbances and, in some cases, products that provide better performance than systems designed in an ad hoc manner. Control methods were essential in distributed software development and E-commerce applications [17]. The progress made in this area indicates that there is considerable value in applying control theory to computing systems.

The control engineer needs to understand how to use discrete-time models and methods within a control and systems context versus continuous-time models and methods in other contexts. Kadrich [18] argues that "security is about control" and recommends to define the network as a control problem. It becomes necessary to apply basic engineering control principles to our networks to control how risk is introduced such that we can predict how our networks are going to react when they are attacked [18].

Controlling congestion in networks requires revision and use of controllers that regulate traffic in the nodes using control theory. Better control schemes could identify the changes of the payload during transmission caused by malware or errors due to malfunctioning equipment [19].

Control theory plays an important role in developing new computing systems, especially complex software systems. The development of new systems require applying a wide range of control principles and methods: optimization of production scheduling, load allocation, optimization of services, decision making under uncertainty with increased levels of decision support, quality of service, and security of systems and data [6].

7) Risk management

In assessing the risk for control systems, use of general methods for risk analysis including specific conditions and characteristics of a control system need to be applied. Process control methods can be used to build a new model. All the devices on the network have a role, and that role can be associated with some form of control. By using this new model, we can more accurately set an acceptable limit of risk, build trust, and thus protect our networks more effectively [18].

Another issue in managing the risk is reducing the vulnerabilities and causes of the vulnerabilities. General models for managing risk through its various phases are available for IT systems [20], [21]. However, methods for risk management that are based on automated tools [22], and intelligent techniques are more beneficial to control systems because they require minimum or no human intervention in controlling the processes [23]. Another important issue is applying vulnerability management life cycle that offers guidance on design and operational processes and technologies needed to find and remediate security weaknesses before they are exploited. It is imperative to analyze risk as a function of asset value, threat and vulnerability. New concepts to analyze the threats and vulnerabilities have to be applied regularly and uniformly. Specific methods for security risk analysis are emerging.

These methods are based on combining concepts of vulnerability tree analysis, fault tree analysis, attack tree analysis, and the cause-consequence [24].

8) Employer's needs

Employers require competencies such as [1]:

- Adopt methods, techniques, tools, and practices of the evolved Systems Engineering
- Acknowledge the increased complexity and adaptability required for the solution of systems and the standards that they adopt
- Understand the possibilities opened by competency development.

Based on the above issues (1-8), we summarize the requirements of new knowledge and skills as the following:

- Information Security
- Networking
- WEB Technologies
- Mobile and Wireless Technologies
- Intelligent Control Systems
- Management
- Systems Design
- Emerging technologies.

This results in updating the programs that prepare the future developers of control systems software. In the following section, we discuss the programs that support new competencies for control engineers. We compare the knowledge areas for disciplines that relate more closely to control systems developers. Also, we present a proposal incorporating new approaches for educating and preparing future control engineers and software developers with adequate skills for design and development of secure, user friendly, and effective control systems.

EDUCATING FOR THE FUTURE

Developers of control systems employ professionals educated in Engineering (most popular are Computer Engineering and Electrical Engineering) and Computer Science programs. New programs in Systems Engineering and Software Engineering prepare professionals for the development of computing systems and software. These programs are based on a curriculum and standards supported by different organizations. We identified Systems Engineering programs offered at the graduate level in many universities world wide across of US, Canada, Australia, Europe, China, etc. Usually, these programs are managed by Computer Engineering departments. However, there are many differences in the core courses and specialized courses taught, project topics, and methods of teaching in Systems Engineering programs [25]. Also, the number of established Software Engineering programs is growing. These programs are offered at the undergraduate and graduate level. The Software Engineering programs are offered by Software Engineering department, or other departments such as Computer Engineering and Computer Science. In the US, the official engineering accreditation board, ABET, has accredited 13 undergraduate Software Engineering programs since 2003, and in Canada, CEAB has accredited nine such programs. Professional certifications in software are also supported in many countries. Studies indicate that there is a knowledge gap on the topics taught in the Software Engineering degrees offered in the universities and skills needed in the job [26]. In the academia, some topics are sometimes underemphasized or overemphasized [26].

Curriculum guidelines and accreditation standards have been set for undergraduate Software Engineering programs. The IEEE Computer Society approved the Software Engineering Body of Knowledge (SWEBOK) 2.0 in 2004 which was adopted as an ISO/IEC Technical Reference in 2005 [27]. The knowledge areas include the following: software requirements, software design, software construction, software testing, software maintenance, software configuration management, software engineering management, software engineering process, software engineering tools and methods, and software quality (see also Table I). The related disciplines to Software Engineering include Systems Engineering, Computer Engineering, Computer Science, Management, Mathematics, Project Management, Quality Management, Software Ergonomics, Information Systems, Information Sciences, Information Technology.

Systems Engineering knowledge plays a key role in the development of computing systems. The International Council on Systems Engineering (INCOSE) states that "Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem: operations performance, test, manufacturing, cost and schedule, training and support and disposal." [1]. Systems Engineers are the primary interface between management, customers, suppliers, and other specialties such as Computer Engineering, Electrical Engineering, Chemical Engineering, Software Engineering, Civil Engineering, Aeronautical Engineering, Environmental Engineering, Reliability Engineering, Safety Engineering, Maintainability Engineering, Manufacturing Engineering, Sales and Marketing, Human Factors Engineering, Electronics, Computer Science, Information Sciences. Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets user needs. The Systems Engineering Body of Knowledge areas (KA-SYE) are documented by the International Council on Systems Engineering (INCOSE) as follows[1]:

1. Business Processes and Operational Assessment (BPOA)
2. System/Solution/Test Architecture (SSTA)
3. Life Cycle Cost & Cost-Benefit Analysis (LCC & CBA)
4. Serviceability / Logistics (S/L)
5. Modeling, Simulation, & Analysis (MS&A)

6. Management: Risk, Configuration, Baseline (Mgt).

We performed an analysis of courses and topics taught in relevant programs for preparing skills in control systems. We identified that knowledge areas are covered in depth or introductory level within different programs providing the education of software professionals involved in the development of control systems. Table I shows the mapping of main knowledge areas of Software Engineering (KA-SE) that are taught in Computer Engineering (CE), Computer Science (CS), Software Engineering (SE), and Systems Engineering (SYE) programs. Based on the data shown in this table, we determined that knowledge areas for Software Engineering are taught differently in other programs as follows:

- approximately 80% coverage in SYE programs
- approximately 60% coverage in CE programs
- approximately 50% coverage in CS programs.

Table II shows the mapping of main knowledge areas of Systems Engineering (KA-SYE) that are taught in Computer Engineering (CE), Computer Science (CS), Software Engineering (SE), and Systems Engineering (SYE) programs Based on the data shown in this table, we determined that knowledge areas for Systems Engineering are taught differently in other programs as follows:

- approximately 70% coverage in CE programs
- approximately 50% coverage in SE programs
- approximately 30% coverage in CS programs.

We performed the mapping of knowledge areas defined in Computing Curricula 2001 for CE and CS programs. We determined that knowledge areas for CE programs are less than 50% covered in CS programs. Although the courses in Information Security are not mandatory knowledge in CS and CE programs, we identified that these courses are more often taught in CS programs.

TABLE I
SE KNOWLEDGE IN RELATED PROGRAMS

KA-SE	CE	CS	SE	SYE
Requirements	x	x	x	x
Design	x	x	x	x
Construction	x	x	x	x
Testing	x	x	x	x
Maintenance			x	
Configuration			x	x
Management			x	x
Process	x		x	x
Tools		x	x	
Quality	x		x	x

TABLE II
SYE KNOWLEDGE IN RELATED PROGRAMS

KA-SYE	CE	CS	SE	SYE
BPOA	x	x	x	x
SSTA	x	x	x	x
LCC&CBA				x
S/L	x			x
MS&A	x			x
MGt	x		x	x

We determined that knowledge in the areas of Information Security, Intelligent Control Systems, Networking, WEB Technologies, Mobile and Wireless Technologies, Management, Systems Design, and Emerging Technologies support new competencies for the developers of control systems to meet the needs of current and future computing systems. Given the main requirements and characteristics of control systems software, we propose that new knowledge areas should be supported by CE and SYE programs. Data depicted in tables I and II shows that CE programs support more broad knowledge in Systems Engineering and Software Engineering than CS programs. Higher education institutions could improve the topics of courses in Computer Engineering and Systems Engineering to include the new knowledge areas. It is a challenging task to convey the basic ideas of systems and control to students in CS who lack the necessary background in mathematics, physics, and engineering. Systems and control is a difficult subject both conceptually and technically [28]. The difficulties of developing and teaching courses in control theory to computer science students in a few universities in US (University of California at Berkeley, Columbia University) are documented. Table III shows the mapping of these knowledge areas. We marked with “X” and “x” a knowledge area that require to be covered in depth or introductory level within these programs.

TABLE III
PROPOSAL NEW KNOWLEDGE IN CE & SYE

KA	CE	SYE
Information Security	X	X
Networking	X	X
WEB Technologies	x	x
Mobile and Wireless Technologies	X	x
Intelligent Control Systems	X	x
Management	x	X
Systems Design	X	X
Emerging Technologies	X	x

CONCLUSION

Datamonitor, a market research firm, predicted growth of revenue from sales of control systems software with a rate from 3.5 to 4 percent per year through 2009 [29]. The expansion of complexity and use of control systems will lead to an increased need for professionals to design and develop new software requirements.

The implementation of improved solutions demands enhancements in the education of software developers. The success depends on a few factors. First, it depends on how researchers and academia envision future realizations. Second, improving skills for control systems developers requires new knowledge in more areas. Third, efforts to develop licensing requirements, curricula, or training programs for developers of control systems should consider the experience of the practitioners who actually perform the work.

REFERENCES

[1] INCOSE, www.incose.org.

[2] E.J. Byres and M. Franz, "Finding the Security Holes before the Hackers Do Vulnerability Discovery in Industrial Control Systems," *ISA Technical Conference, Instrumentation Systems and Automation Society*, Chicago, October, 2005, [Online]. Available: http://www.byressecurity.com/pages/publications/technical-papers/.

[3] E.J. Byres, D. Hoffman, and N. Kube, "On Shaky Ground – A Study of Security Vulnerabilities in Control Protocols,", *5th American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology, American Nuclear Society*, Albuquerque, NM, November 2006, [Online]. Available: http://www.byressecurity.com/pages/publications/technical-papers/.

[4] A.M. Wildberger, "AI & Simulation," SIMULATION, Vol. 73, No.1, 1999, pp. 55-56.

[5] R. Sanz, and K.E. Arzen, "Trends in Software and Control," *IEEE Control Systems Magazine*, vol. 23, no. 3, 2003, pp. 12-15.

[6] Conference Reports, "Four Focused Forums", *IEEE Control Systems Magazine*, vol. 26, no. 4, 2006, pp 93-98.

[7] S.M. Rinaldi, J.P., Peerenboom, and T.K. Kelly, "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies," *IEEE Control Systems Magazine*, vol. 21, no. 6, 2001, pp. 11-25.

[8] W.R. Dunn, "Designing Safety-Critical Computer Systems," *IEEE Computer*, vol. 36, no. 11, 2003, pp. 40-46.

[9] R. Cummings, "The Evolution of Information Assurance," *IEEE Computer*, vol. 35, no. 12, 2002, pp. 65-72.

[10] M. Fisher, "Protecting Binary Executables," *Embedded Systems Programming*, vol. 13, no. 2, 2000, pp. 24-30.

[11] M. Hentea, M. "Information security management" In M. Pagani, (Ed.), *Encyclopedia of Multimedia Technology and Networking*. Hershey, Pennsylvania, IDEA GROUP REFERENCE, 2005, pp. 390-395.

[12 K. Hangos, R. Lakner, and M. Gerzson, *Intelligent Control Systems: An Introduction with Examples*. Kluwer Academic Publishers, 2001.

[13] M. Hentea, "Intelligent System for Information Security Management: Architecture and Design Issues," *Journal of Issues in Informing Science and Information Technology,2007*, accepted.

[14] M.S. Obaidat, "Performance Evaluation of Computer and Telecommunications Systems, " SIMULATION, Vol. 72, No.5, 1999, pp. 55-56.

[15] W.J. Davis, X. Chen, X., A. Brook, and F.A. Awad, "Implementing On-Line Simulation with the World Wide Web," SIMULATION, Vol. 73, No.1, 1999, pp. 40-54.

[16] B.S. Heck, L.M. Wills, and G.J. Vatchtsevanos, "Software Technology for Implementing Reusable, Distributed Control Systems," *IEEE Control Systems Magazine*, vol. 23, no. 2, 2003, pp. 21-35.

[17] J.L. Hellerstein, Y. Diao, S. Parekh, and D.M. Tilbury, "Control Engineering for Computing Systems, *IEEE Control Systems Magazine*, vol. 25, no. 6,2005, pp. 56-68.

[18] M. Kadrich, "Something is Missing," *Computer Security Journal*, vol. XXII, no. 4, 2006, pp. 1-16.

[19] M. Welzel, *Netwotk Congestion Control managing internet traffic*. John Wiley & Sons, Ltd., Hoboken, NJ, 2005.

[20] R. Craft, G. Wyss, R. Vandewart, and D. Funkhouser, "An Open Framework for Risk Management," *1998, 21st National Information Systems Security Conference Proceedings*, [Online]. Available: http://csrc.nist.gov/nissc/1998/proceedings/paperE6.pdf

[21] H.E. Glavin, "A Risk Modeling Methodology," *Computer Security Journal*, vol. XIX, no. 3, 2003, pp. 1-29.

[22] W. Ozier, "A Framework for an Automated Risk Assessment Tool," 1999, [Online]. Available: http://www.theiia.org/itaudit/index.cfm?fuseaction=forum&fid=228.

[23] M. Hentea, "Enhancing Information Security Risk Management with Data Mining and Fuzzy Logic Techniques," *Proceedings of 19th International Conference on Computer Applications in Industry and Engineering*, Las Vegas, Nevada, 2006, pp. 132-139.

[24] S. Vidalis and A. Jones, "Using Vulnerability Trees for Decision Making in Threat Assessment," [On-Line] Available: http://www.glam.ac.uk/socschool/research/publications/technical/CS-03-2.pdf.

[25] D.M. Buede, "Existing Academic Activity in Support of Systems Engineering," *Academic Forum 2001 INCOSE Symposium*, July 2001, [On-Line] Available: http://www.incose.org/symp2001/forum/buede%20preso.ppt.

[26] T.C. Lethbridge, "What Knowledge is Important to a Software Professional?" *IEEE Computer*, May 2000, pp. 44-50.

[27] SWEBOK, [On-Line] Available: http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf.

[28] People in Control, *IEEE Control Systems Magazine*, vol. 27, no. 2, 2007, pp 14-19.

[29] D. Geer, "Security of Critical Control Systems Sparks Concern," *IEEE Computer*, vol. 39, no. 1, pp. 21-23, 2006.