How Do You Create an Internet of Things Workforce?

Joanna F. DeFranco
Penn State University

Mohamad Kassab
Penn State University

Jeffrey Voas
IEEE Fellow

Editor:
Jeffrey Voas, NIST;
j.voas@ieee.org

Internet of Things (IoT) products and cyber-physical systems (CPS) are being utilized in almost every discipline. According to *Forbes*, there will be a significant increase in spending on the design and development of IoT applications and analytics. Furthermore, the most significant increase in spending will be in the business-to-business (B2B) IoT systems (such as manufacturing, transportation, and utilities), which is projected to reach $267 billion by 2020.[1] In addition to B2B, smart products are becoming more prevalent, such as thermostats, energy monitors, and light bulbs. Products that sense, learn, and react to user preferences are gaining popularity.

There are also CPS/IoT applications for healthcare with the goal of improving a patient's treatment regime. For example, the closed-loop insulin delivery system connecting a glucose monitor to an insulin pump can continuously alter the amount of insulin dosed to a patient to assist in managing the patient's blood sugar. In fact, any product that continuously monitors patient activity to improve treatment would be an effective IoT application. Imagine how much more effective treatment could be for a Parkinson's patient when a physician has more than a static snapshot from an office visit exam. With months of data and information, the physician could determine a more effective treatment plan. Accordingly, engineers and computer scientists also need the appropriate training to build safe and effective systems, whether part of the IoT or not. However, it is not sufficient to simply add one or two IoT or CPS courses to an existing program curriculum for students to gain the knowledge necessary to build reliable, efficient, and safe CPS or IoT systems.

It is time for a new engineering discipline that adapts to the reasons why IoT and CPS are different than existing engineering disciplines. History has shown that new engineering disciplines follow the newest technologies, and IoT and CPS are the newest technology trends. Electrical engineering emerged in the late 19th century with the invention of the electric motor. Chemical engineering emerged during the Industrial Revolution with the mass production of chemicals. Biomedical engineering rolled out in the early 1980s. Even the latest engineering discipline—software engineering—emerged as a result of the increased complexity of software systems. Now, with the capability of "things" that collect, aggregate, calculate, and send mounds of data for actuation, we argue that it is time for a new engineering/computer science discipline to emerge that is focused on this space.

 2469-7087/18/$33.00

A college-level program to educate a new workforce with the necessary skills to build effective and safe IoT and CPS systems is warranted. We suggest developing CPS and IoT engineering programs at colleges and universities that have established engineering departments, given the estimate of needing hundreds of thousands of IoT-educated engineers in the near future.[2] This does not suggest that vocational schools and other educational institutions cannot also help build this needed workforce—all help is needed.

A search on Indeed.com for US-based jobs that mentioned IoT resulted in more than 1,900 job opportunities. This doesn't include open positions in data analytics testing, algorithms, machine learning, or security, which are important disciplines in the design and implementation of CPS/IoT. In fact, the Bureau of Labor and Statistics predicts a 30 percent increase in jobs related to those technical domains by 2026.

IOT/CPS TRAINING

Academic institutions might already be considering an IoT-focused computer science degree or adapting curriculum from existing programs. To assist with that effort, co-author Voas, along with Phillip Laplante, mapped out five "Network of Things" (NoT) primitives that have been discussed by the National Institute of Standards and Technology (NIST)[4] relative to IEEE/ACM's 2013 computer science curricula knowledge areas (KAs; see Table 1).[5]

NoT is a term that applies to both CPS and IoT. The five primitives of all NoT systems include sensors (something that measures physical properties, such as RFID), aggregators (software to transform data from a sensor), a communication channel (data transmission, such as wired or wireless), an eUtility (software or hardware to execute processes, such as a database), and a decision trigger (which creates the final result, such as an actuator). Note that any specifically purposed NoT might not include all five. For example, some NoTs don't have sensors.

The easiest way to think about this is that the "things" are what make IoT unique. Many people question whether IoT is just marketing hype or if there is a science behind it. So, what is IoT? We'd better know before we start educating people about it.

IoT is an acronym of three letters. "I" (Internet) existed long before the acronym was termed and "o" does not matter, so "T" (things) is the letter in the acronym that we should pay attention to. So, the five NoT primitives define the "Lego-like" building blocks for any IoT-based system. The primitives are the "things," and this is where we need to focus our education efforts.

There are 18 KAs in computer science (for example, architecture and operating systems) that correspond well with understanding IoT in terms of the "things." Voas and Laplante recommended a set of topics to consider when creating new curricula or when modifying existing computer science curricula.[3] Further, if you are looking more at CPS issues than IoT concerns, modifying a systems engineering, electrical engineering, or mechanical engineering curricula might be worth pursing as well.

Table 1. IEEE/ACM computer science knowledge areas.[5]

1	Algorithms and complexity	10	Networking and communications
2	Architecture and organization	11	Operating systems
3	Computational science	12	Platform-based development
4	Discrete structures	13	Parallel and distributed computing
5	Graphics and visualization	14	Programing languages
6	Human-computer interaction	15	Software development fundamentals

7	Information assurance and security	16	Software engineering
8	Information management	17	Systems fundamentals
9	Intelligent systems	18	Social issues and professional practice

CPS/IOT PROGRAM STATUS

We reviewed CPS/IoT-related programs at the top 50 universities ranked by Collegechoice.net (an aggregate of *US News & World Report* and the National Center for Education Statistics) and TopUniversities.com (international universities) for IoT and CPS course offerings as of December 2017. More than half of those universities (see Table 2) had courses with a CPS/IoT focus, most of which were in graduate programs. Interestingly, more than half of those courses are taught in electrical engineering and computer engineering programs.

Table 2. Number of Internet of Things (IoT)/cyber-physical systems (CPS) courses at the top 50 ranked universities.

Universities with IoT/CPS courses	Total IoT/CPS courses	Undergrad courses	Graduate courses
28	49	17	32

The course descriptions of those 49 courses reveal that the IoT primitives are covered; however, only 11 percent seem to cover all five primitives (see Figure 1). These courses, "Interconnected Embedded Systems," "Networked Cyber-Physical Systems," "Internet of Things—Intelligent and Connected Systems," and "Body Sensor Networks in the Internet of Things," appear to be introductions to IoT and CPS technical and design understanding.

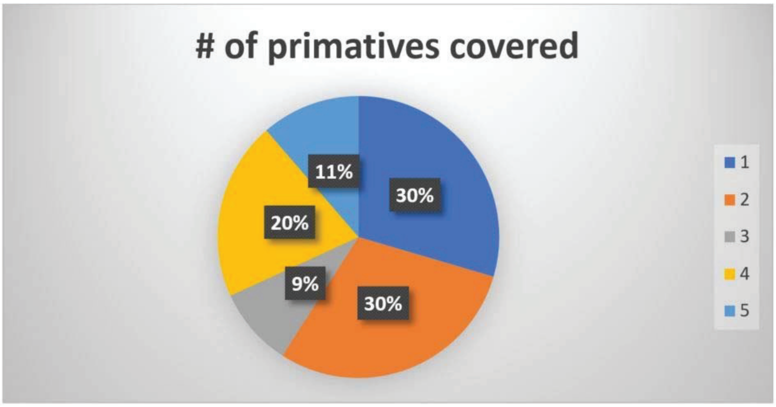

Figure 1. Number of IoT primitives covered in the courses at the top 50 ranked universities.

We reviewed more extensively some of the courses in CPS/IoT to gain a deeper understanding of the course content and structure. The courses reviewed focused on embedded systems with a CPS concentration, and on either CPS or IoT specifically. The difficulty in creating these courses is addressing the challenges of a CPS/IoT designer,[6] which include heterogeneous network technology integration, fault tolerance on the many individual devices in a system, prioritizing critical actions during system degradation situations, and distributed system energy management.

Despite these challenges, there are many open opportunities for course development. In addition to course content, there are numerous projects and use cases to include in these courses that would foster real-world experiences for students. For example, creating a "smart city" could involve installing sensor boxes around a city to monitor pedestrian or automobile flow to make informed decisions about traffic lights, bus stops, or where the next convenience store is located. A robot project might also be an effective learning tool by using camera and location sensors on the robot and then communicating the information (such as the location of the robot in the room) back to an embedded server. Another interesting project might be simulating a miniature factory by using a robotic arm and a controller to communicate to the outside world, where an app is developed to control the arm. These sample projects could be integrated into CPS or IoT courses.

Table 3 shows plausible examples of CPS or IoT course content. These examples are also mapped to the IEEE/ACM KAs discussed earlier (see Table 1). Considering that these could be CPS/IoT courses, they will also map to the five NoT primitives.

Table 3. Example CPS/IoT courses.

Course focus	Knowledge areas covered
Embedded systems with more focus on critical thinking about the effects of embedded software on the behavior, safety, and reliability of a CPS	1, 2, 3, 4, 10, 11, 12, 14, 15, 17
Implementation of functional prototype sensor/control networks (wired or wireless through available mobile device apps)	2, 3, 8, 9, 10, 12, 13, 16, 17
Learning to design embedded and CPS systems with real-time behaviors	1, 3, 4, 9, 10, 11, 12, 16
CPS applications focusing on resource management, timing constraints, distributed sensing, computation, control, modeling verification, and testing	1, 8, 9, 10, 11, 12, 13, 16, 17,
Embedded controls, field programmable gate array design, and server programming	1, 2, 8, 9, 10, 12, 14, 15, 16
CPS architecture and their vulnerabilities to cyber-attacks	1, 2, 3, 6, 7, 9, 10,12,14,17
IoT physical and logical architecture and functional blocks, communications protocols, smart objects, security, data analytics, system management, and ethical and environmental impact	1, 2, 3, 6, 7, 8, 9, 10, 12, 16, 17, 18
Focus on IoT by using RaaS (Robot as a Service) integrating a robot, sensors, and actuators into a cloud computing environment	1, 2, 3, 4, 5, 6, 9, 11, 12, 13, 14, 15, 16, 17
Focus on IoT to design and prototype an ambient intelligence system	2, 6, 9, 10, 12, 13, 14, 15, 16
A lab implementing the functionality of an entire facility to test specific concepts	2, 8, 9, 10, 11, 12

RECOMMENDATIONS

Elective courses are the least onerous way to begin the development of a CPS/IoT curricula. It is a challenge to revise an existing academic program and even more challenging to offer a new academic program—especially one that is to be accredited.

This article originally appeared in
IT Professional, *vol. 20, no. 4, 2018.*

In Table 3, we described examples of elective courses that could combine concepts from the major academic programs (for example, electrical engineering or computer science), as well as integrating CPS/IoT concepts into the course content. However, this is not a long-term solution given the workforce needs for employees with expertise in developing safe, reliable, and secure CPS/IoT systems. In other words, elective courses are a quick and easy way to pioneer a program and gauge interest among current and future students; however, for a long-term solution, academic institutions need to begin to define new curriculums and degrees. One option would be certificate programs that eventually grow into full undergraduate or graduate degree programs.

Another approach would be to integrate CPS/IoT concepts into existing program courses by developing learning modules. These modules would highlight specific CPS/IoT concepts. For example, researchers at Virginia Tech created CPS security-focused learning modules.[7] The easily accessible learning modules contain a specific learning objective with tools and hands-on exercises relevant for conventional embedded systems, control system design, and cybersecurity courses.

There is no doubt that a complete CPS/IoT curriculum will require core skills from multiple existing engineering and/or computer science programs. Courses on subjects such as embedded systems, computer security, software architecture, software construction, and others will apply. Accordingly, the path of least resistance is to create new programs by modifying existing programs, as many of these courses might already exist at the institution and only require slight modifications. This process can be used with other academic program KAs. This appears to be the most efficient way to create new CPS/IoT educational programs that are relevant and timely.

REFERENCES

1. L. Columbus, "Internet of Things Market to Reach $267B by 2020," *Forbes*, 29 January 2017; www.forbes.com/sites/louiscolumbus/2017/01/29/internet-of-things-market-to-reach-267b-by-2020/#f2ddc5609bd6.
2. S. Krishnamurthi, "More Education Needed to Realize the Internet of Everything," *Network World*, 10 December 2014; www.networkworld.com/article/2857890/internet-of-things/more-education-needed-to-realize-the-internet-of-everything.html.
3. J. Voas and P. Laplante, "Curriculum Considerations for the Internet of Things," *Computer*, vol. 50, no. 1, 2017, pp. 72–75.
4. J. Voas, *Networks of 'Things'*, report SP 800-185, NIST, 2016; doi.org/http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-183.pdf.
5. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, ACM and IEEE Computer Society, 2013; www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf.
6. P. Gonzalez-Nalda et al., "Building a CPS as an Educational Challenge," *Int'l J. Online Eng.*, vol. 10, no. 4, 2014, pp. 52–58.
7. P.P. Deshmukh, C.D. Patterson, and W.T. Baumann, "A Hands-on Modular Laboratory Environment to Foster Learning in Control System Security," *IEEE Frontiers in Education Conference* (FIE), 2016; doi.org/10.1109/FIE.2016.7757669.

ABOUT THE AUTHORS

Joanna F. DeFranco is an assistant professor of software engineering at Penn State University. Contact her at jfd104@psu.edu.

Mohamad Kassab is an associate research professor of software engineering at Penn State University. Contact him at muk36@psu.edu.

Jeffrey Voas is an IEEE Fellow. Contact him at j.voas@ieee.org.