A Practical Introduction to Engineering a Secure Internet of Things

James Solderitsch
Villanova University
Department of Electrical and Computer Engineering
800 East Lancaster Ave,
Villanova PA 19085
james.j.solderitsch@villanova.edu

ABSTRACT

Everyone is talking about the Internet of Things (IOT). When focused on an industrial setting, a common alternate name is the Industrial Internet: machines talking to machines (M2M) autonomously, with people often reduced to secondary roles. There are big challenges ahead, not the least of which is how to make this Industrial Internet safe and secure. This paper focuses on a description of these challenges and how engineering education can adapt to offer students the means to understand the challenges and give them the tools and techniques to engineer a secure IOT. Part of the educational process must focus on the things being enabled with network connectivity. We present an interesting development ecosystem based on devices from Texas Instruments and a device programming and monitoring framework built on the open source Contiki operating system. In addition we discuss an application development environment that is being made available in an educational program offered by the PTC Corporation, which recently acquired the IOT company ThingWorx. We present some of the benefits that students and faculty can expect to receive from combining both of these capabilities into a cohesive instructional program.

Note: An early draft of this paper and was created while the author was a member of the Cyber Security R&D group within Accenture Technology Laboratories that was supporting a recently established company-wide strategic innovation initiative on the Industrial Internet.

Categories and Subject Descriptors

K.6.5 [**Computing Milieux**] Security and Protection – *invasive software, physical security, unauthorized access*; H.3.4 [**Information Systems**]: Systems and Software – *current awareness systems, distributed systems, information networks;* C.2.3 [**Communication Networks**]: Network Operations – *network management, network monitoring;* D.2.8 [**Software Engineering**]: Metrics – *process metrics, performance measures*

General Terms

Management, Measurement, Reliability, Development, Security

Keywords

Security, Devices, Protocols, Analytics, Cloud, Continuous Monitoring, Open Source, Internet of Things, Trust

1. INTRODUCTION

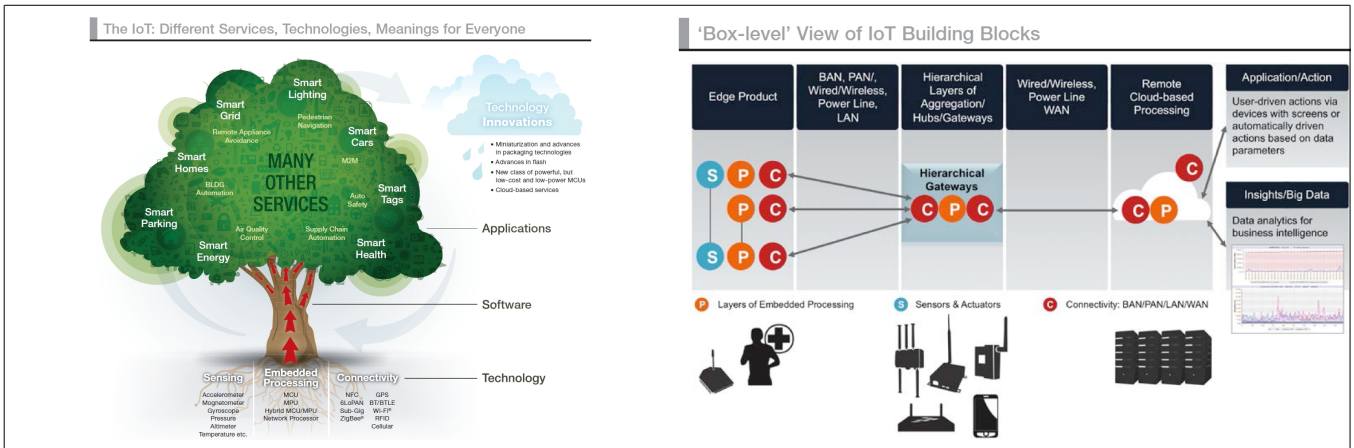
Just use "Internet of Things (IOT)" as a search term in your favorite browser and you will be both inundated with information and get a sense that this is a very hot area for Information Technology (IT) and Operations Technology (OT) in general, crossing over from consumer-oriented applications like the connected home (e.g. the Nest Thermostat) into the industrial space with connected construction vehicle fleets and intelligent sensor-driven manufacturing operations to name just two areas. Karimi and Atkinson provide two nice summary diagrams in [1] that are reproduced in Figure 1. The messaging in these diagrams is pretty clear: *smart* requires lots of infrastructure to achieve it and the IOT has three main areas to consider: Sensing, Processing, and Communicating. All of these provide extensive security challenges and working in the area of secure IOT engineering means you must consider all of these aspects. You need to have all of them in front of you as you consider device development (including sensors and actuators), device management, device communication, data protection across the IOT space, and the offering of new forms of security monitoring embedded in the back-end, and probably cloud-based, operations control center.

In this paper we want to acknowledge the significant challenges that need to be understood to create a safe and secure world of smart things and smart systems. We want to raise awareness in the context of an engineering student's educational program of how you can essentially experience the Internet of Things first hand.

2. MAKING THE INTERNET OF THINGS SECURE – BEYOND IT/OT CONVERGENCE

The rapid expansion of the Internet of Things has opened a Pandora's Box from a security perspective for industries and consumers alike ([3], [4] and [5]). As smart technology and business models evolve to address the needs of customers, enterprises need to understand and address security and privacy throughout the lifecycle. Previously secure and closed systems can now be accessed and controlled remotely, potentially enabling hostile parties to take control of our appliances, machines, finances, and most worryingly, our identities.

For the IOT to reach its expected presence and influence in modern society, providers and consumers need to believe that the benefit of the service being provided is higher than the risk of using it. From an industrial perspective, utility providers want to optimize their operations to deliver uninterrupted services safely (e.g. pipeline sensors, controls, UAVs to monitor pipeline fields). If these services are disrupted, they can impact safety as well as brand reputation. Customers will embrace a more connected lifestyle only if they feel protected from unwarranted intrusions into their personal space, and can be given a clear understanding



(a) IOT Complexity (b) Sensing, Processing and Communicating

Figure 1: What is the IOT? (Karimi [1])

of what personal data is being shared and the received benefits of sharing it.

Connected devices are becoming prominent among consumers within their homes, vehicles and even exercise equipment (e.g. Nike FitBit). Consumers want to reap the benefits from connected devices to optimize their energy consumption at home, to help lose a few pounds or simply make their lives easier. However, recent reports suggest that several thousand everyday consumer devices have already been "hacked" including televisions, routers/home networking devices, and even appliances, which will impact the brand reputation for the retailer and the manufacturers of the products.

To be successful, the IOT requires product designers and engineers, operations teams and security organizations to collaborate to fully understand the impact of consequential risks and unforeseen threats to their business and the services that they provide. New and agile security thinking is required to keep our lives defended against those with malicious intent.

We believe that an experiential, immersive approach for students learning how to build IOT capabilities will help bring these generic areas of functionality and security to the forefront. An educational environment is required where these capabilities can be touched, examined and manipulated to understand how IOT computing presents significant challenges as a fusion of both information technology and operational technology.

3. IOT CAPABILITY DOMAINS

The IOT/Industrial Internet Capability Model shown in Figure 2 identifies five capability domains that will be the basis for unlocking new productivity gains and efficiencies for the IOT that unifies Operational and Information Technology. This model was collaboratively created while the author was working in the Cyber Security R&D group within Accenture Technology Laboratories.

Security is shown as a vertical domain in the figure that is crosscutting through all of the other domains. Security has to be built in across the ever evolving operational and information technology landscape, especially from an end-to-end application security or software assurance perspective as these applications and processes are the very fabric for how the connected enterprise and the envisioned connected world will evolve.

Each of the other domains highlight important areas in which research and development activities are rapidly evolving to provide breakthroughs whose collective combination will empower the IOT. These domains include:

- **Sensor-Driven Computing –** A new application architecture for integrating hundreds to thousands to millions of sensors and their data streams; e.g. connected products, homes, cars, factories and cities.
- **Industrial Analytics –** Industrial software, systems and controls that are enabled by continuous innovation in how data is moved and stored; the use of analytics and the cloud is widely appreciated for traditional IT systems but these will assume a much more prominent role where the IOT is a fusion of IT and OT leading to orders of magnitude increases in the volume, variety and velocity of data (truly "Big Data") and the distributed processing elements to derive consumer and business value from the data.
- **Machine-Aware Processes –** People, devices and processes that are united and operate intelligently and eventually support advanced forms of decision making and reasoning (e.g. robotics, autonomous vehicles, artificial intelligence, machine learning).
- **Context-Aware Experiences –** Seamless collaboration between human and machine based on shared up-to-date awareness of the current environment; e.g. device and operator location, conditions such as weather or the state of the electrical power grid, supply chain state, recent operational trends and even new kinds of wearable technology like Google glass or other wearable devices.

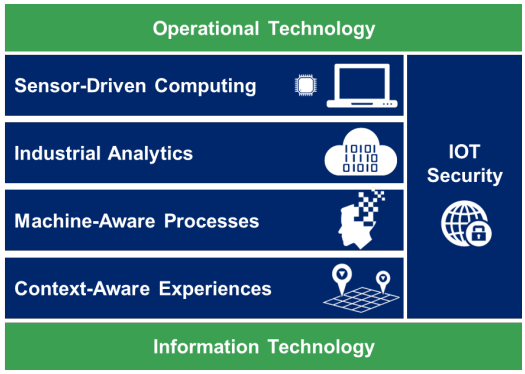


Figure 2: IOT Capability Model

4. OPEN SOURCE ENVIRONMENTS AND AFFORDABLE DEVICES

To truly experience the Internet of Things from devices at the edge to powerful analytical engines in the cloud that process data from a myriad of devices and convert it to operation management and decision-making, you need more than the device part of the experience. Many device candidates exist including Raspberry Pi's and Arduinos, but these primarily offer tinkering experiences at the device level only. Development environments are somewhat tricky to set up and use effectively and you are on your own to figure out how to manage a suite of these devices and create an effective operations center concept to support them.

Contiki [6] is an Open Source platform [7] wrapped in a downloadable Virtual Machine that has led to the creation of an Internet of Things company called Thingsquare. While Thingsquare is not the only company that offers developers the opportunity the buy/build devices and enable and test wireless communication among them – others include Ayla and Axeda – Contiki devices have been created with security principles baked in [8] including secure encrypted communications between devices and an industrial strength approach to firmware updates. Having begun in late 2013, Thingsquare continues to evolve a cloud-based platform supporting teams of developers and researchers working in a distributed manner on device development, deployment and monitoring with an API (not yet well-documented unfortunately) to create cloud-based services of their own. Researchers have the ability to participate directly in the development of the open source project including the provision of advanced security features.

5. CONTIKI THINGSQUARE CLOUD PLATFORM

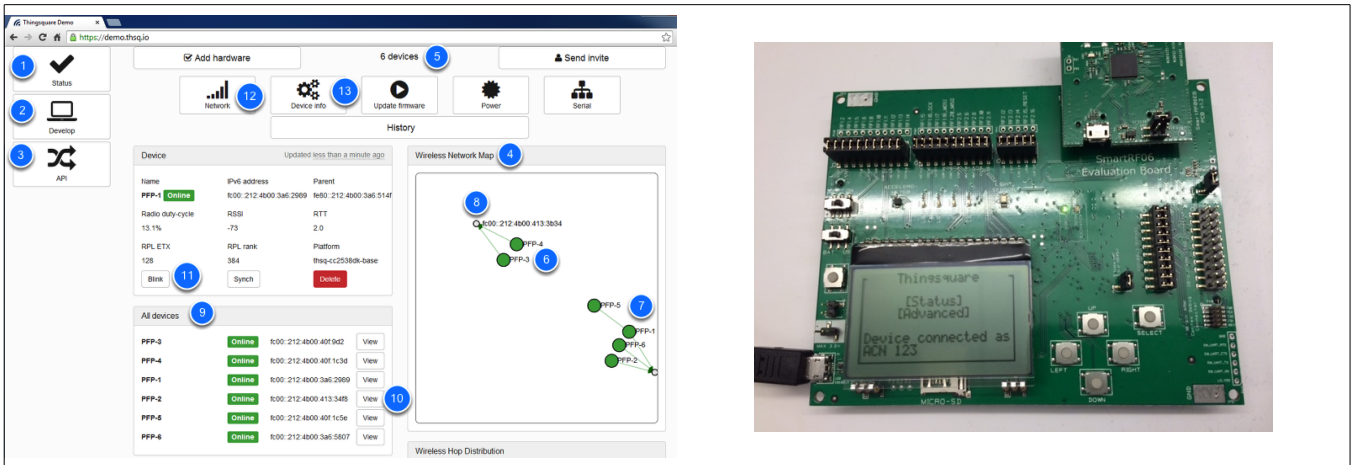
Figure 3(a) shows the current (fall 2014) Thingsquare platform status screen that offers a complete development and monitoring environment for inexpensive Thingsquare compatible devices manufactured by Texas Instruments. These devices feature onboard sensors, displays and 6Lowpan enabled wireless radios. A picture of one of the devices is shown in Figure 3(b).

If a device has been connected successfully, it will show a connected message on the display board's status LCD as shown in Figure 3(b). During the rest of this section, we will describe some of the features available under the Status, Develop, and API (no screenshot) options. Note that this demo cloud site is also functional on smart devices with an adequate web browser such as smart phones and tablets including iPhone, iPad and Android devices.

All three of the items on the left in Figure 3(a): Status (1), Develop (2) and API (3) provide opportunities to interact with the devices. The Wireless Network Map (4) shows the currently registered devices. As shown, there are 6 devices registered (5) – clicking the Status icon will get you back here from other parts of the demo UI. As shown in the Network Map, there are two clusters of devices. The smaller one (6) was actually running on the author's home network and the larger one (7) was operational in the Accenture Tech Labs office at the time this screenshot was made. The small grey node (8) is an internet gateway identified by its IPv6 address. The actual devices appear as green nodes when on-line (they will be grayed-out when off-line) and their status is summarized in the All devices panel on the left (9). You can select a device for closer examination by clicking the View button (10) or by clicking a green node in the Network Map. In the above picture, we are looking at the PFP-1 device in the Device panel.

The bulk of the information in the Device panel is technical information about the state of the device. You can test connectivity to the device from this interface by clicking the Blink button (11). This will cause the LEDs on the selected device to blink once. This is the default way of viewing Device status – clicking the Network icon (12) will return to this view. Clicking the Device Info (13) icon allows more fine-grained examination for all of the deployed devices and offers the ability to set and inspect device variables including LED state and sensor readings. The Texas Instruments devices have on-board accelerometer and light level sensors. This detailed view is shown in Figure 4(a) where we are looking at the device PFP-2.

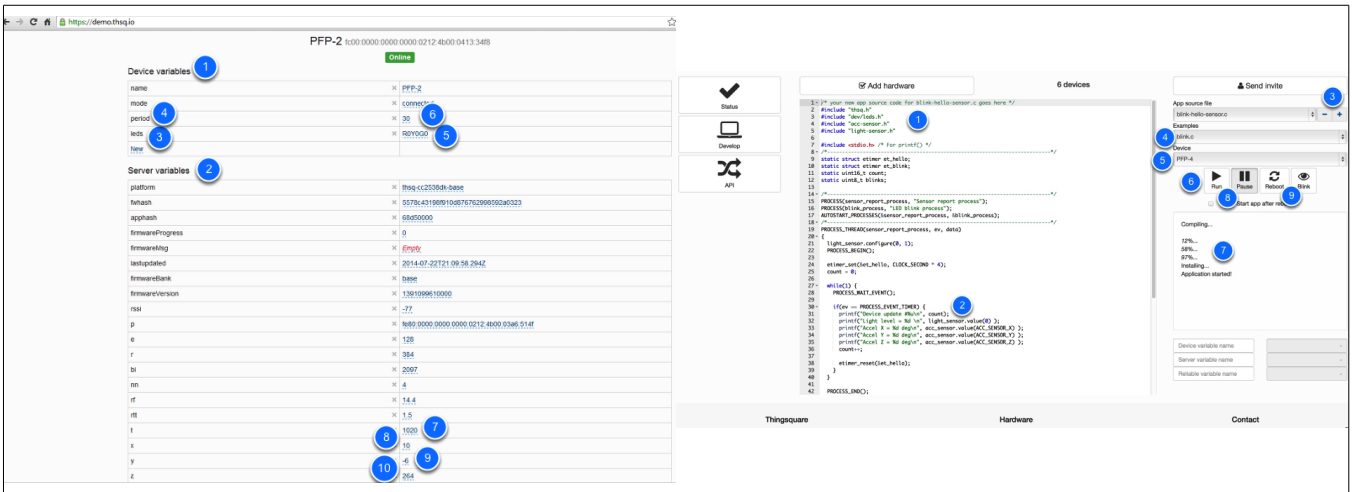
The Device Info view has been scrolled down to expose many of PFP-2's variables. The actual variables defined for a device will vary based on the firmware version and past usage. From here all device variables can be inspected and interacted with. Device variables are broken down into the Device (1) and Server (2) categories. Server variables are those managed by the Thingsquare demo cloud back end and result from inspecting the state of each device. Of particular note are the leds (3) and the period (4) device variables. The former (5) conveys the current led status and the latter (6) is the server refresh cycle time (shown here as 30 seconds) where the available variables are polled to refresh the values in this device variable table. R0Y0G0 means that the leds



(a) Thingsquare Device Status Summary Panel

(b) Thingsquare Device with LCD Display

Figure 3: Full Spectrum IOT



(a) Device Details Page (b) Develop: Cloud Programming Interface

Figure 4: Thingsquare Capability Demo Examples

are off, whereas R100Y100G100 sets the brightness for the Red, Yellow and Green leds to that level. These variables are also settable from this display. You can set the variables by clicking on the value (5) and changing the RYG string value there: setting to R100Y100G100 turns the leds on to level 100. You can also change the refresh period time by clicking and editing the number.

The TI board light level and accelerometer sensor values are shown as the values of t (7), x (8), y (9) and z (10). The values shown in the Device panel will refresh according to the set period time. If you handle the board – for example pick it up and turn it over – you will see changed values here because the light levels and accelerometer axis orientation readings have changed since the last refresh.

Through the cloud interface, more than one person can be logged in and interacting with registered devices simultaneously. In fact, access is managed through a team definition and a supervisor such as a faculty member can invite students to join her team. Note that the leds variable, even though identified as a Device variable, reports the current status of led brightness with each refresh cycle. If another team member is logged into the demo site and changes the led settings for a particular device, the leds variable will report the changed brightness next time the Device Info page refreshes. As such this page serves as a crude Human Machine Interface (HMI) for the device where you can both set brightness as well as perceive externally caused changes in brightness. You can also sense accelerometer or light sensor changes if a device in a remote location is being handled.

Although we do not include more screenshots in this paper for device status, the following is a summary of other features available in the Status portion of the Thingsquare platform cloud Figure 3(a):

- **Update firmware:** inspection and the ability to push firmware updates securely using AES encryption and checksum verification either to individual devices or to the entire set of devices.
- **Power:** shows more detailed information of the power modes of the set of registered devices and their connectedness to the device network.
- **Serial:** used for debugging devices through the USB/Serial connection available for each device including the ability to push string data from the cloud UI back to an individual device; on Windows the PuTTY tool can be used to establish and monitor a device serial connection when a USB cable is connected between a device and a computer running PuTTY.

The Develop portion of the cloud UI is shown in Figure 4(b). This part was introduced early in 2014 and provides an improved mechanism to make software changes to a device's firmware. Previously the recommended procedure was to run the Contiki OS development environment in a Virtual machine (VM) and physically connect the devices to the VM and update the firmware via a device-specific firmware flasher. This method of pushing new software to a device is still available but is quite cumbersome to use. In the cloud Develop interface we have a mini-IDE including a source code editor, a file picker, a device selector, and a compilation status panel showing the interaction for the Develop UI with the selected device.

The demo program shown here (1) echoes using the printf string (2) current light and accelerometer sensor readings to the USB serial port for the device and blinks the device leds (this part of the source program not shown). Source code for several demos is available in a device code source Zip file available from Thingsquare. This particular demo was adapted by the author of the paper that extends a simpler example contained in the Zip file to combine device led control along with the programmatic reporting of device sensor values. You can add your own demo by clicking the + button (3). There are 3 sample device programs available from the Examples selector (4). You pick which device you want to use to run the program from the Device selector (5) – PFP-4 is shown. You can run the program by clicking the Run button (6). Compilation status, including error reporting, and application activity is shown (7).

To push a program to the device, click the Run button and this causes a Compile (in the cloud), an Upload and Install (to the device) and finally the execution of the program. You can interrupt the running program with the Pause button (8). To restore the device to its previous state, you use the Reboot button (9). All of this capability is managed through the cloud UI.

Much more complicated programming examples are possible on the devices including the use of websockets, a feature of HTML 5.0, to push device sensor and operational status values to your own server rather than the cloud server run by Thingsquare. In fact, the demo.thsq.io server [9] uses websockets under its covers to obtain and report device variable values as shown in Figure 4(a).

The API features of the Thingsquare Demo cloud (item 3 in Figure 3(a)) are still in their initial stage of development. API documentation has not yet been released as of the writing of this paper, but Thingsquare has published plans to offer customer-specific cloud instances where use of the API will be crucial to creating customized applications. For now, the API UI (not shown) provides the means to generate an API key and some illustrative examples of how to use this key to make REST-style calls to your devices using a command line tool such as curl or wget. You can both read and change device variable values using these calls and the API page provides examples of how to do this.

6. THINGWORX APPLICATION DEVELOPMENT PLATFORM

The cloud UI provided by Thingsquare is able to demonstrate IOT concepts, especially at the device level, but does not function as a customizable or configurable application development environment. In fact the UI can change without warning as we have experienced several times during the course of our experimentation with Contiki, the TI devices and the available cloud backend. If we want students to be able to create applications quickly and easily and experience the power of the IOT, the current demo environment is limited.

Thingsquare is planning to offer a paid-for service, using Amazon Web Services as a delivery mechanism, for users to create their own cloud-based applications. But this capability is not yet operational and it is not clear how much educational support Thingsquare will be willing to provide. Rather than waiting for the realization of this service, we wanted to see if another IOT technology might be useful from the application-building side.

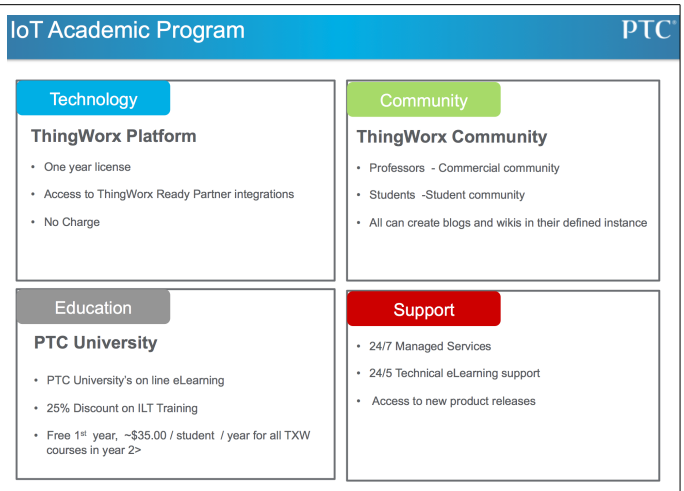
The PTC Company is well known in engineering education circles and provides cost-effective means for students to use commercial grade software at little to no cost. They are the distributor of the MathCAD design software for example. Recently PTC has acquired the company ThingWorx (based in Exton PA, nearby to Villanova) because of its first class IOT application development platform. Very recently PTC has also acquired the Axeda Company mentioned earlier. Despite the name similarity, ThingWorx is completely unrelated to Thingsquare but the application creation features offered by the ThingWorx platform have a natural fit with the kinds of devices we have worked with.

As stated in the ThingWorx Academic Program FAQ: *ThingWorx is a rapid application enablement platform that allows companies to develop purpose-built Internet of Things applications much faster than traditional software tools, resulting in rapidly developed solutions that solve real business challenges. Applications built on ThingWorx can bring value to smart connected products.* A demonstration video of ThingWorx in action is available [10].

PTC is just beginning to roll out its academic program for educational use of ThingWorx and we have been provided with an early access release. As shown in Figure 5(a), there are a number of distinct components included in the platform. Figure 5(b) shows the planned educational rollout of the ThingWorx educational framework. One of the examples provided in the educator's tutorial package uses a Raspberry Pi device as the means through which sensor data collected by the Raspberry Pi is communicated to a simple application developed using ThingWorx. From an IOT device perspective, a Raspberry Pi device is too much like a real computer, running a version of the Linux operating system. We believe that the TI devices are more representative of realistic "edge" devices that send data into a centralized collection and management point.

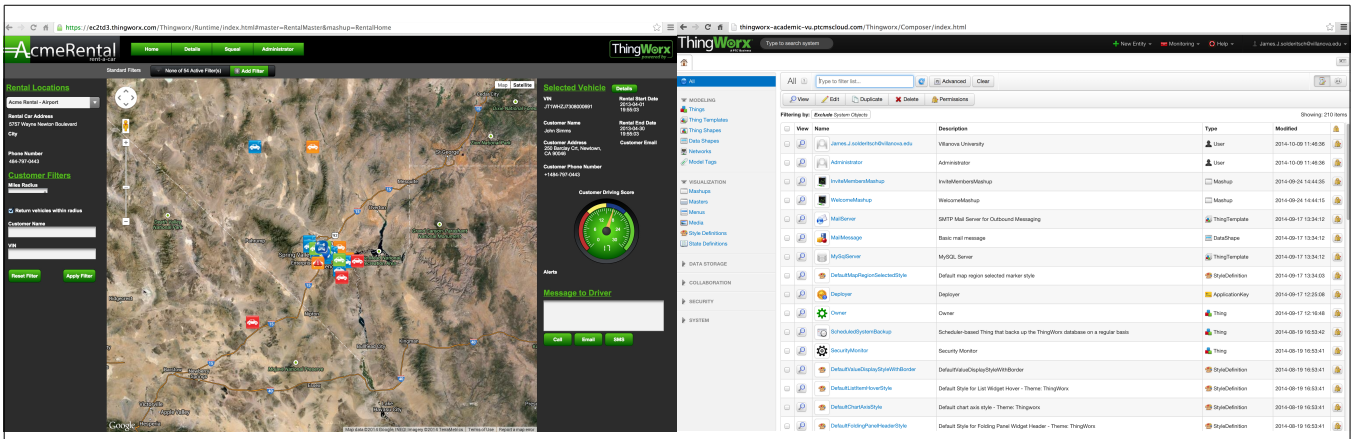
Figure 6(a) shows a Rental Vehicle management application, the kind of application that can be built using ThingWorx. With this application you can track vehicles "live" by their location, get detailed reporting about an individual vehicle and even communicate live with the driver of a vehicle. The developer of an IOT application is guided by a powerful UI that minimizes the amount of code that must be written by hand. The ThingWorx Composer UI is illustrated in Figure 6(b). You use the composer to model the environment in which your "things" operate along with the requisite business logic that allows your application to derive value from your network of things. You can create visualization "widgets" using the Mashup Builder to show the data being collected from your things. ThingWorx also offers its own efficient event driven storage and can also interface to external Big Data capabilities built on Hadoop and similar platforms. Your applications can include collaboration widgets that allow users to communicate with one another directly within the application. Some of these features are visible in the application screen shown in Figure 6(a).

(a) Application Creation Components



(b) Educational Program

Figure 5: ThingWorx Introduction



(a) Sample ThingWorx Application (b) ThingWorx Composer

Figure 6: ThingWorx Operations

We are convinced that engineering students will be best served in their introduction to the Internet of Things by a two-pronged approach featuring an open source framework for devices and their programming (such as provided by Contiki, which is supported by Thingsquare) and an easy-to-use application development environment (such as that provided by ThingWorx) that enables students to build powerful and appealing applications.

7. CONCLUSION

Beth Stackpole in a recent article in the online publication Desktop Engineering [11] presents four important skills for engineering the IOT: **Embedded software, Communication capabilities, Instrumentation,** and **Data and security**. We think that both of the Thingsquare and PTC/ThingWorx platforms and others like them are the future for IOT awareness, training, and engineering and the means for students to acquire these skills. Many writers are producing articles on the inherent dangers arising from the lack of built-in security when IOT use cases and applications are being created in large numbers (e.g. Bruce Schneier [12] and a recent GIGAOM opinion piece [13]). But you can't create a new security model or approach in a vacuum without hands-on experience. This paper is an invitation to educators to roll up your sleeves and get started. The end-to-end suite of devices, tooling, cloud-based monitoring and application development provides a complete realization of the Internet of Things and provides a test bed for device experimentation including many aspects of security.

8. ACKNOWLEDGEMENTS

We wish to credit Adam Dunkels (Thingsquare CEO and creator of Contiki) and the Thingsquare team for help and support and useful background information about Contiki and the development kit. Recent conversations with Andy Barlow from PTC/ThingWorx have also been extremely helpful and Andy has made available an educational usage license for the ThingWorx application development environment. We also want to acknowledge a chat with Kaivan Karimi at the October 2013 IOT/M2M symposium in Washington DC and recommend his IOT summary in [1] as a clear presentation of the present and future of the IOT including some important aspects of security. We also thank Accenture colleagues Martin Glowik and Walid Negm for reviews of the initial paper draft and suggestions and discussions that led to the development of the IOT capability model.

9. REFERENCES

[1] Kaivan Karimi and Gary Atkinson, What the Internet of Things (IoT) needs to become a reality, Freescale and ARM White Paper, June 2013, DOI= http://www.freescale.com/files/32bit/doc/white_paper/INTOTHNGSWP.pdf

[2] Earl Perkins, Definition: Operational Technology Security, Gartner Report, September 2013.

[3] Rodrigo Roman, Pablo Najera, and Javier Lopez, Securing the Internet of Things, *IEEE Computer*, Volume: 44, Issue: 9, September 2011, pp. 51-58.

[4] Huansheng Ning, Hong Liu and Laurence T. Yang, Cyberentity Security in the Internet of Things, *IEEE Computer*, Volume: 46 , Issue: 4, April 2013, pp 46-53.

[5] Securing the Internet of Things: New Ways to Deploy Trust in Enterprise Computing Beyond the PC, *Trusted Computing Group*, Webcast and Slides, DOI= http://www.trustedcomputinggroup.org/resources/securing_the_internet_of_things_new_ways_to_deploy_trust_in_enterprise_computing_beyond_the_pc.

[6] Contiki, Wikipedia, DOI= http://en.wikipedia.org/wiki/Contiki .

[7] Contiki Open Source Project, DOI= http://www.contiki-os.org/.

[8] Lander Casado and Philippas Tsigas, ContikiSec: A Secure Network Layer for Wireless Sensor Networks under the Contiki Operating System, *NordSec '09 Proceedings of the 14th Nordic Conference on Secure IT Systems: Identity and Privacy in the Internet Age*, pp. 133-146, DOI= http://www.cse.chalmers.se/~tsigas/papers/ContikiSec_Nordsec.pdf.

[9] Thingsquare Cloud Demo, DOI= https://demo.thsq.io/.

[10] ThingWorx Demonstration Video, DOI= http://support.ptc.com/apps/corporate-videos/video/Corporate-Videos/-2123317325/PTC-and-ThingWorx-Solution-Demonstration/2990418211001

[11] Beth Stackpole, 4 Skills for the Internet of Things, *Desktop Engineering*, August 2014, DOI= http://www.nxtbook.com/nxtbooks/level5/desktopengineering_201408/index.php?bm=normal#/16.

[12] Bruce Schneier, The Internet of Things is Wildly Insecure – And Often Unpatchable, January 2014, DOI= https://www.schneier.com/essay-468.html.

[13] Stacey Higginbotham, The internet of things needs a new security model. Which one will win?, *GIGAOM*, January 2014, DOI= http://gigaom.com/2014/01/22/the-internet-of-things-needs-a-new-security-model-which-one-will-win