5-2017

Designing for Reuse in an Industrial Internet of Things Monitoring Application

Ethan T. McGee
Clemson University

Matthew Krugh
Clemson University

John D. McGregor
Clemson University

Laine Mears
Clemson University, mears@clemson.edu

Recommended Citation

McGee, Ethan T.; Krugh, Matthew; McGregor, John D.; and Mears, Laine, "Designing for Reuse in an Industrial Internet of Things Monitoring Application" (2017). *Publications*. 81.
https://tigerprints.clemson.edu/auto_eng_pub/81

Designing for Reuse in an Industrial Internet of Things Monitoring Application

Ethan T. McGee
Clemson University
School of Computing
Clemson, SC USA
etmcgee@clemson.edu

Matthew Krugh
Clemson University
Automotive Engineering
Greenville, SC USA
mkrugh@clemson.edu

John D. McGregor
Clemson University
School of Computing
Clemson, SC USA
johnmc@clemson.edu

Laine Mears
Clemson University
Automotive Engineering
Greenville, SC USA
mears@clemson.edu

ABSTRACT

The Internet of Things (IoT) continues to experience rapid growth, and its influence is extending into previously unreached domains. However, some of these new domains impose specific limitations that complicate the design and implementation of IoT systems. Examples of such limitations are the exclusion of specific protocols, restrictions on the types of data that can be collected, requirements about what information can be transmitted to the public and controls around how that communication occurs. Capturing, representing and designing for these limitations as well as reuse is essential for the quick and successful deployment of such projects. In this paper, we present a case study of an IoT human in the loop monitoring system built for use within an industrial setting. We report our experiences with both designing the first deployment of the system as well as designing variation points into the software architecture to account for future iterations and deployment into other environments.

Keywords

Architecture Analysis & Design Language; Human Issues; Intellectual Property

1. INTRODUCTION

The Internet of Things (IoT) is an increasingly popular paradigm that promises to allow a variety of "things" (refrigerators, microwaves, thermostats, vehicles, etc.) to be augmented with networking and sensing capabilities, enabling them to work together towards accomplishing a common set of goals [9]. This connectivity allows these everyday devices to become "smart" [16], and also promises the synthesis of separate data streams allowing for better understanding and reasoning [9]. However, in domains with a heavy human presence, one of these data streams might be the actions undertaken by the human to affect the common goals and outcomes. Such systems are also known as "human in the loop." In such cases, it is necessary to integrate the human as an element of the smart system, tracking their actions and possibly providing a means to suggest modifications to their behavior.

These visions of an interconnected web of smart devices have been explored in many domains, especially in the home [10], transportation [17] and medicine [14]. However, not all domains are equally accepting of the benefits and infrastructure of the IoT. Manufacturing facilities are one such instance. Many of the protocols common in the IoT, such as Bluetooth and Bluetooth Low Energy (BLE), interfere with machines present in a manufacturing process. Additionally, the openness of the protocols forming the backbone of the IoT creates new attack vectors against facilities protecting trade secrets. As such, IoT system development in these contexts has strict limitations imposed that complicate the design and implementation of IoT systems. And for systems designed for use in multiple organizations, each facility can and often does impose unique restrictions.

In developing a smart manufacturing system with a heavy human presence, there are also limitations that are imposed on the system(s) used to integrate the human into the smart system. From the company's perspective, the devices worn by the employee cannot impede or interfere with the manufacturing process. From the employee's perspective, the device should protect their right to privacy while still providing enough information to the company to be useful.

In order to be first to market, the development schedules for IoT systems are often shortened with little room for error. Ensuring that these smart systems are developed successfully, accounting for all limitations, requires the design to be as accurate as possible since the majority (approx. 70%) of errors are introduced in the design / requirements phases of the Software Development Life Cycle [13]. It is also necessary, especially for systems having multiple deployments with differing requirements for each deployment, to reuse components of the software system as much as possible.

In this paper, we present a case study reporting our experi-

WASHES '17, May 29, 2017, Salvador, Brazil

ence modeling and designing an industrial sensing system in which humans are an integral part of the system. We record our experiences designing the system, accounting for the various restrictions of each organization for which a deployment of the system is planned. We also give our experiences implementing a version of the system for the first organization which accounts for that organization's specific limitations. In section 2, we provide background information concerning Software Product Lines and Bluetooth Low Energy. Then, in section 3, we explain the manufacturing process of the first organization and the restrictions imposed by that organization. Finally, in section 4, we provide our design and analysis of the sensing system. Due to privacy concerns, the companies for which this system was developed will remain anonymous.

2. BACKGROUND

We now provide information necessary for understanding this work. We first introduce and provide an example of Software Product Lines (SPLs). We also provide a brief overview of the Bluetooth Low Energy (BLE) protocol.

2.1 Software Product Lines

SPLs are a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and are developed from a common set of core assets in a prescribed way [12]. SPLs have achieved remarkable benefits including productivity gains, increased agility, increased product quality, mass customization, and improvements in other business drivers [11].

Each SPL generates a suite of products, and each product has a common set of features, otherwise known as a core asset base. These common, shared features are choreographed within a common architecture, or family architecture, that is also shared by each product in the product line. However, the family architecture of the product line contains abstract points for each product to implement individually, also known as variation points. Every product in the product line will provide a unique implementation, or variant, to these variation points whereby each product endows itself with a set of distinct features not available in the core asset base.

SPLs are of particular importance to the IoT. They allow companies to maintain a common sets of assets that can be reused across multiple products lowering development effort and time. They also allow for better customization of products which can be tailored individually to the needs of customers increasing desirability. An example of this can be seen in the IoT prototyping platform LaunchPad produced by Texas Instruments. LaunchPad offers a wide variety of development kits built around a common set of compatible processors with different attached sensors and peripherals depending on the sensing and communication needs of the user. Texas Instruments also sells customized boards if a pre-made development kit does not fit the particular needs of a user.

2.2 Bluetooth Low Energy

Bluetooth Low Energy (BLE) is a wireless communication technology designed to provide many of the capabilities of traditional Bluetooth while offering reduced cost and power consumption [19]. BLE is a widely supported protocol found among many of the IoT devices available on the market today. Devices such as smart watches, health wearables and wireless headphones utilize the technology to communicate with other devices such as mobile phones or computers. Most operating systems, mobile and desktop, provide at least some level of support of the protocol, either natively or through libraries that can be installed to enable support. In addition to wide compatibility, BLE features a range of approximately 100 meters; however, the presence of interference can reduce this range considerably. It also features relatively low latency in the presence of interference free transmission, and it offers the capability of service discovery [19].

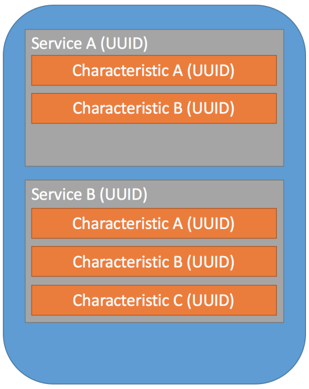

Figure 1: Bluetooth Low Energy GATT Diagram

BLE uses the Generic Attribute Protocol (GATT) as its primary means of transmission. GATT provides a set of services, each exposed by a name and Universally Unique IDentifier (UUID), as is shown in figure 1. The service represents a set of related values (known in GATT as characteristics), each exposed with their own individual UUID. Each characteristic is read from and written to using the associated UUID to address into the characteristic database, an internal memory store that holds the values of characteristics during operation. Some UUIDs are reserved for specific peripherals such as heart monitors, etc. by the standards committee overseeing the BLE protocol [1].

GATT characteristics have several properties that define how they operate, further giving the user control over access and power consumption. For example, a characteristic can be read-only, write-only or read-write. They can also be active or passive. A "passive" characteristic requires that a connected device poll the characteristic for updates on a periodic basis as the connected device will not be notified when an update is available. An "active" characteristic will send a notification that an update has occurred along with the new value. Note, however, that the connected device is not required to subscribe to notifications, and the use of notifications can increase the power consumption of the BLE device.

3. MANUFACTURING PROCESS

Now we examine the IoT sensing system and its role in the organizations for which it was designed. The IoT sensing system was designed for use within multiple organizations, each imposing different restrictions on the system.

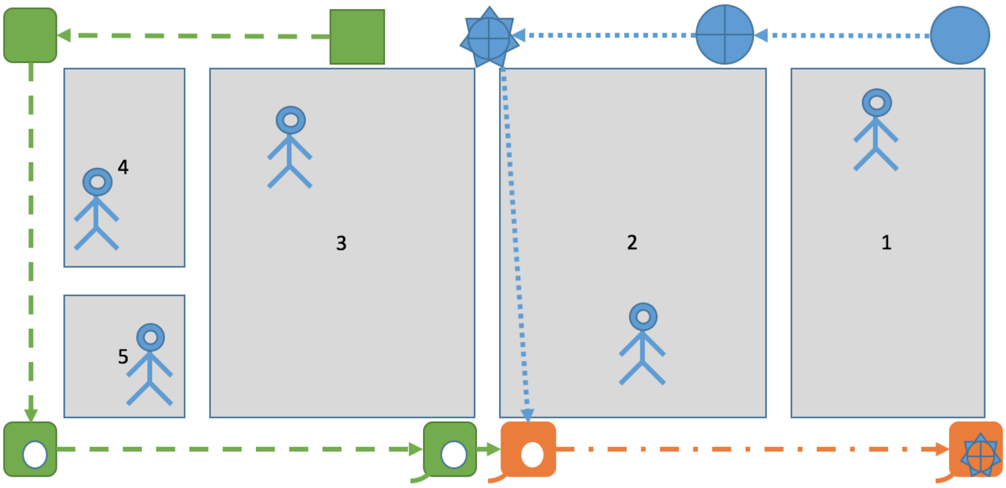

Figure 2: Assembly Line Outline

The product line was designed to be deployed in each of these organizations, but as of this writing, it has only been deployed to one. The system is designed to augment the existing assembly processes of each facility providing more information about the assembly process, especially the interaction between human and machine. We now provide some background information about the assembly process of this organization, hereafter referred to as Company A.

Company A produces transmission control units for major international automobile manufacturers. The assembly lines, for which this instantiation of the system was developed, are diagrammed in figure 2. The line is divided into subsections (represented by the dotted, dashed and dotted / dashed sections of figure 2), at the start of which a raw product or set of finished products is introduced. As the materials / products move down the line the materials are converted into finished products (represented by the dashed and dotted subsections) or the finished products are assembled together (represented by the dotted / dashed subsection).

Each subsection consists of several stations, and each station performs a set of tasks on the product loaded into the machine by workers. Example actions include the drilling of holes, soldering electronic control units and welding pieces in place. Once a machine completes its task, the employee responsible for the machine removes the part and passes it down the line to the next station. Unlike some assembly facilities where workers operate a single station, employees of this facility operate *loops* of machines (represented by the gray areas of figure 2). Each is a literal elliptical section of the assembly area lined on two sides by machinery. These areas are not always limited to a single subsection of the assembly line, and workers are responsible for maintaining the machines on both sides.

As is represented by the dotted arrow of figure 2, a part does not necessarily move all the way around the assembly line. Once employees of loop 1 and 2 move a part through the dotted subsection, the finished part is moved to a bin at the start of the dotted / dashed subsection where it is paired with pieces from the dashed subsection.

The device developed is to be worn by employees during a normal shift. It tracks the employees location within the assembly line throughout their shift while simultaneous tracking movements. This data is aggregated to determine if there are steps that could be taken by improve the overall throughput of the assembly line.

4. SENSING SYSTEM DESIGN & IMPLEMENTATION

We now provide details concerning the requirements, design and implementation of the system. We will also provide an overview of some of the analyses run against the system.

4.1 Requirements / Design

Each organization for which a deployment of the system was planned imposed unique requirements on the system, however, most features of the system were shared among all organizations. These shared features specified the assets that form the core asset base of our SPL and are discussed below.

- **A. Wireless sensors worn by the employees should communicate only to base stations assigned to the employee's line.** - Since there is a possibility of multiple lines within the same facility being recorded at the same time, a method of preventing cross-over recordings was needed.

- **B. The XDK platform should be used as the sensing system, and the base station should consist of 1 or more Raspberry Pi 3s.** - The XDK system [6] is a sensing platform that had been used successfully in many of the organizations involved. Since these organizations were already using the system, getting approval to use this system in their environment was not necessary. Additionally, organizations not using the XDK were more accepting of the platform since other companies in the same domain were using it.

  The same was true of the Raspberry Pi 3 [2]. Many of the organizations involved in this effort were already using the Pi in other projects so their approval was not needed. Of the organizations not using the Pi, gaining approval to use the system was easier because other organizations within the same domain were using it.

- **C. Location readings should occur approximately every 5 seconds.** - Employees on the lines remain in one place for upwards of 30 seconds while performing their assigned tasks. Sampling every 5 seconds would allow energy conservation while still sampling at a high enough frequency to accurately capture location at a moment in time.
- **D. Battery powered devices should be capable of lasting through one 10 hour shift.** - Most shifts, among the organizations involved, last 8 hours, however, some of the organizations use 10 hour shifts.

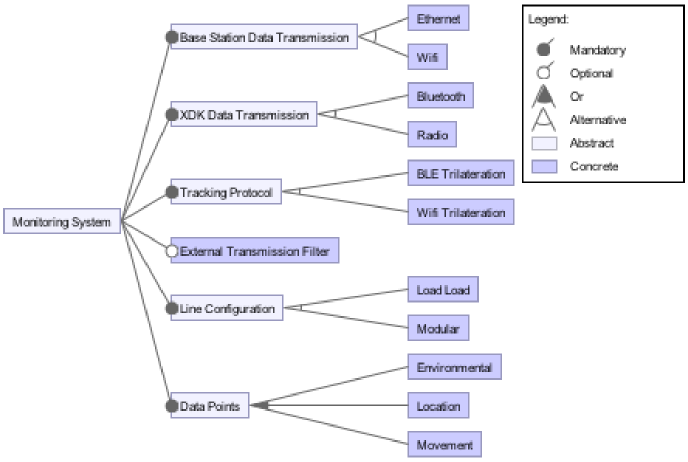

Figure 3: Feature Model

For the remaining features, those not shared by all organizations, we created variation points in the architecture to account for the differences. These variation points and their variants are presented in a feature model in figure 3. We now further discuss these options.

Each person on the assembly line was fitted with an XDK unit that would track the attributes needed by their organization. However, each organization used various assembly styles, and the style had to be encoded into the system so that tracking of data points would be accurate. Every organization wanted to track employee location, however, some organizations also wanted to track movements such as stooping to pick up a tool. Other organizations requested that we track environmental data such as the facility temperature at that location on the line. Sampling rates are adjusted based on organization need. Each XDK transmits its data to a line base station via either BLE or Radio. BLE was not permitted in some organizations due to its interference with existing line machinery.

After the base station had aggregated data from the XDK units, it would transmit the aggregated data to a central location for long term storage and analysis. Some organizations had existing Wifi infrastructure that could be leveraged to transmit this data, others required the use of ethernet connections. Transmission rates were determined based on organization need. Finally, some organizations gave permission for us to transmit data from the central storage location to Clemson for additional analysis. Of these organizations, some required a filter that stripped certain data elements before transmission.

4.2 System Design for Company A

Figure 4 shows the instrumented assembly line for company A. It was decided that the base stations for the system would be a set of 5 Raspberry Pi 3 computers placed directly around the assembly line as the XDK units would be using BLE communication. Each Pi was located roughly at the start of a loop within the assembly line.

For the XDK units (requirement B), version 1.6.0 of the development platform was used to develop software for the devices to be worn by employees. Each XDK unit was assigned a unique id that identified it to the Pis around the assembly line to which the XDK was associated (requirement A). Pis would not know the ids of XDK units from other assembly lines thus preventing cross-readings. Each Pi would establish a connection to one or more XDK units from their line and after the BLE pairing process completed, a signal would be sent to each XDK alerting the XDK to start the sampling process.

As company A requested readings for all types of data, both accelerometers, both gyroscopes, the magnetometer, the environmental sensor and the light intensity sensor on the XDK were enabled. The accelerometers, gyroscopes and magnetometer were sampled every 100 milliseconds (10 hertz) with the reading being transmitted immediately upon sampling (requirement C). The environmental and light intensity sensors were sampled every second (1 hertz) with the reading also being transmitted immediately upon sampling (requirement D).

For central storage and analysis, a Microsoft SQL Server 2012 database server in the company's IoT infrastructure was used. Data generated from the XDK was stored on the Pi for approximately 1 minute before a copy process would execute transferring the generated Comma Separated Value (CSV) files from the Pi to the database server for upload into the database. Immediate external communication of data over the internet was not permitted. Instead of using our filtering process, Company A elected to have database managers scrub the data before exporting to CSV files which were emailed weekly to Clemson.

4.3 Discussion

Throughout the design process, there existed multiple options for how the system could be developed, even after factoring in requirements that limited decisions such as the requirement that the XDK prototyping platform was required for use. However, as is common, each option provided both benefits and risks that had to be weighed against one another. We now provide an overview of one of these options, its benefits / drawbacks and the analysis used to make the final decision.

4.3.1 Active vs. Passive BLE

Company A requested that BLE communication be used for transmission of data from the XDK units to the base stations. Thus a decision of whether to use active or passive BLE communication (activity diagram of a single transmission of all sensors is given in figures 5 and 6) was needed. At

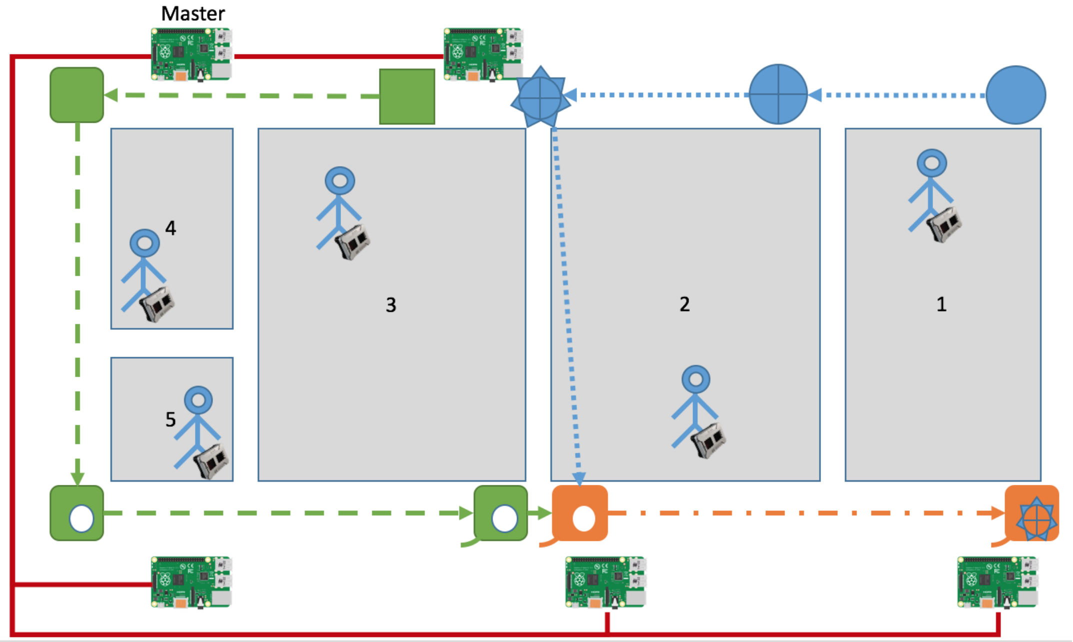

Figure 4: Instrumented Assembly Line Outline

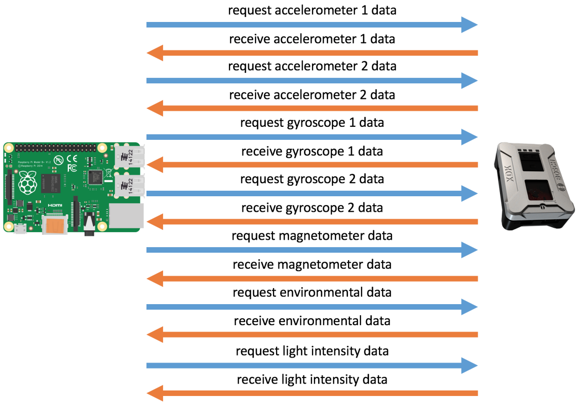

Figure 5: Passive Activity Diagram

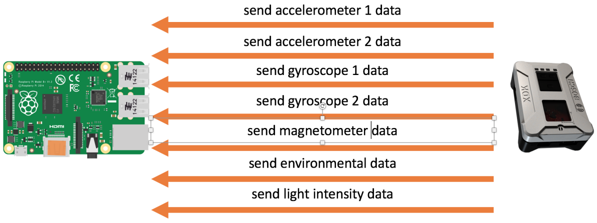

Figure 6: Active Activity Diagram

this point in the design process, no code had been developed for the XDK and neither detailed analysis of the power consumption of the XDK nor analysis of the device's battery life capabilities were available. It was thought that the use of passive communication would allow the device to operate for an entire 10 hour shift (requirement D). However, it was unknown if the high levels of interference present on Company A's assembly line would increase the latency such that passive communication would not be able to meet the sampling rate requirements of Company A. Thus an analysis of both active and passive communication's effects on battery life and throughput were needed.

Working under the assumption that passive communication would allow us to meet our required operating time, we began working to determine if passive communication would meet our data transmission rate thresholds. To determine that passive was indeed the optimal choice for the XDK sensing system, two architectures were constructed, one representing the passive version of the system and the other representing the active version of the system. In both cases, the underlying architecture for the XDK unit remained the same with one difference; BLE notifications were enabled on the active system and they were disabled for the passive version. An overview of the XDK unit's architecture is shown in figure 7.

Having no easy or direct way to perform power draw analysis on the XDK unit, we instead elected to do a simple timing test. A simple version of the base station software was constructed that recorded timestamps of when the base station connected to an XDK device and when the base station disconnected, either due to battery loss or interference. We fully charged the test XDK unit and enabled passive reading. The unit and base station were then placed in a lab that had similar interference to Company A's assembly line. The device and base station were collected the following morning, and the process was repeated with active sending enabled. In passive mode, the device lasted for 28 hours but did not meet our throughput requirements. In

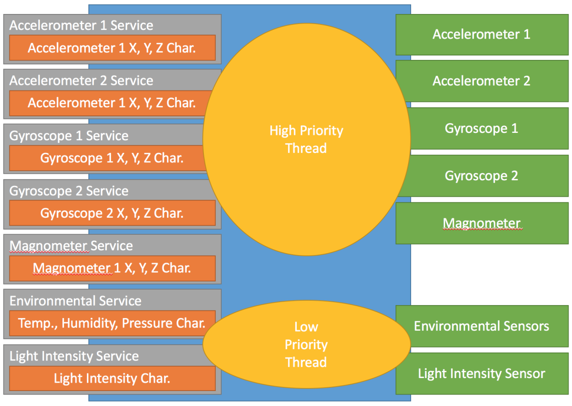

Figure 7: XDK Architecture

active mode, the device lasted for 18 hours and met our throughput requirements. Thus, active BLE was chosen.

4.4 Lessons Learned

During the design process, two primary lessons were learned. First, it is important to ensure that requirements are appropriately prioritized so that if multiple requirements influence a decision, the most important requirements can be assured first, then less important. Second, ensuring that all requirements are fulfilled in the design of a system is imperative. We now discuss the second lesson in more detail.

4.4.1 Missed Requirements

During the final week of system testing, it was discovered that a requirement had been missed in both the design and implementation. This requirement centered around the inclusion of a pressure plate that could be used to determine when the employee had started their loop and was unique to Company A. Fortunately, the addition of a pressure plate proved somewhat trivial, although it was included in a non-desirable way that will have to be re-factored in later revisions of the software. After the system was deployed, a review was conducted to determine if there was a method that could be utilized to assure that a requirement was represented in the design and successfully met.

A package of tools, ALISA, for the Architecture Analysis & Design Language (AADL) provide techniques for requirements specification, verification and assurance. It was added to the project based on the experience missing a requirement. There are four primary file types, each containing a specific type of information, as indicated by the filename extension:

- **.goals** - Goal files contain the high-level goals for the entire project or for specific components of the architecture.

- **.reqspec** - ReqSpec files contain the requirement descriptions (and possibly verification activities for the entire project or for specific components of the architecture.

- **.verify** - Verify files contain the methods for verifying individual requirements, which may require linking with external libraries and programs.

- **.assure** - Assure files tie together verification activities by grouping them under components.

ALISA supports numerous other file types, each containing a different type of supporting information, such as a listing of stakeholders, etc, but those are beyond the scope of this paper.

The use of tools such as ALISA can help prevent missed requirements and they also help ensure that the design produced from the requirements is as accurate as possible. ALISA, although not used at the beginning of the project, has been since integrated and will be used going forward to ensure that all requirements are successfully met.

5. RELATED WORK

A number of "human-in-the-loop" IoT systems have been recently developed [22]. [20] built a system that reduced energy wastage of computers by monitoring the actions of the user, and putting the computer into a lower power state when the user was "distracted". [24] created a system that suggested users move to lower-traffic areas in order to more evenly spread the load on wifi networks. [23, 25, 15] all developed systems that aided impaired users by allowing them to travel to locations [23] or to reach things that they would not be able to normally reach due to their impairment [25, 15]. [8] monitored the current status of a patient connected to a medication pump to determine if it was safe to supply another dose of medication at the next programmed delivery time. [18, 7] proposed methods of more accurately controlling the climate controls of buildings based on current occupancy. In addition, several innovative smartphone applications allow users to crowd source information [5, 4], and other applications attempt to determine a user's actions taken in real life and then translate them into a virtual environment [21, 3].

Our work differs from these groups in that we are primarily focused on manufacturing engineering and producing systems that augment the assembly process. However, the concepts used to build these systems are very much applicable to our system as are the lessons learned from their construction. For example, the method used by [24] to move users to lower load wifi locations is a possibility that could be used in future expansions of the manufacturing system we have developed, particularly expansions focused on providing suggested actions to the user for increasing line productivity. Other works such as [21] offer innovative methods of tracking a user's activity that could be used to enhance or improve our system for determining the user's current activity.

6. CONCLUSION

In this paper, we have examined the design and implementation of an IoT sensing system to be used in multiple organizations. Throughout the design of the system and the implementation of the first deployment, multiple decisions involving complex trade-offs were made. In order to gain better insight into the system and to gather data on the system, a model of the system was constructed and analyzed. Measurements from tests were recorded in the model along with verifications to ensure that the model was able to meet the requirements. Despite the steps taken to ensure an appropriate design was reached, lessons,several of which we have discussed, were still learned that can be applied to future projects.

Future work on this project will include the incorporation of a pattern recognition and feedback system into the IoT sensing system. This will allow the system to detect problems occurring on the assembly line in real time, offering suggestions to the employees, through the use of wear-ables, for how to maintain near-optimal line productivity, if possible. The lessons learned from the design and implementation of this system will be factored into future iterations, as will the models and analyses generated as a result of the existing system's design.

7. REFERENCES

[1] Adafruit. https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt. Accessed: 2017.

[2] Raspberry pi 3 model b. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. Accessed: 2017.

[3] Second life. http://secondlife.com. Accessed: 2017.

[4] Waze. https://www.waze.com. Accessed: 2017.

[5] Weathersignal. http://weathersignal.com. Accessed: 2017.

[6] Xdk. http://xdk.io. Accessed: 2017.

[7] Y. Agarwal, B. Balaji, S. Dutta, R. K. Gupta, and T. Weng. Duty-cycling buildings aggressively: The next frontier in hvac control. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 246–257. IEEE, 2011.

[8] D. Arney, M. Pajic, J. M. Goldman, I. Lee, R. Mangharam, and O. Sokolsky. Toward patient safety in closed-loop medical device systems. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, pages 139–148. ACM, 2010.

[9] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[10] K. Bing, L. Fu, Y. Zhuo, and L. Yanlei. Design of an internet of things-based smart home system. In *Intelligent Control and Information Processing (ICICIP), 2011 2nd International Conference on*, volume 2, pages 921–924. IEEE, 2011.

[11] P. Clements and J. McGregor. Better, faster, cheaper: Pick any three. *Business horizons*, 55(2):201–208, 2012.

[12] P. Clements and L. Northrop. *Software product lines.* Addison-Wesley,, 2002.

[13] P. Feiler, J. Goodenough, A. Gurfinkel, C. Weinstock, and L. Wrage. Four pillars for improving the quality of safety-critical software-reliant systems. Technical report, DTIC Document, 2013.

[14] R. S. Istepanian, A. Sungoor, A. Faisal, and N. Philip. Internet of m-health things'm-iot'. 2011.

[15] D.-J. Kim and A. Behal. Human-in-the-loop control of an assistive robotic arm in unstructured environments for spinal cord injured users. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 285–286. ACM, 2009.

[16] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton. Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, 14(1):44–51, 2010.

[17] D. Kyriazis, T. Varvarigou, D. White, A. Rossi, and J. Cooper. Sustainable smart city iot applications: Heat and electricity management & eco-conscious cruise control for public transportation. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–5. IEEE, 2013.

[18] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse. The smart thermostat: using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 211–224. ACM, 2010.

[19] F. Martelli. Bluetooth® low energy, 2014.

[20] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin. Reducing energy waste for computers by human-in-the-loop control. *IEEE Transactions on Emerging Topics in Computing*, 2(4):448–460, 2014.

[21] M. Musolesi, E. Miluzzo, N. D. Lane, S. B. Eisenman, T. Choudhury, and A. T. Campbell. The second life of a sensor: Integrating real-world experience in virtual worlds using mobile phones. *Technology*, 100:3, 2008.

[22] D. S. Nunes, P. Zhang, and J. S. Silva. A survey on human-in-the-loop applications towards an internet of all. *IEEE Communications Surveys & Tutorials*, 17(2):944–965, 2015.

[23] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir. The future of human-in-the-loop cyber-physical systems. *Computer*, 46(1):36–45, 2013.

[24] R. Schoenen and H. Yanikomeroglu. User-in-the-loop: Spatial and temporal demand shaping for sustainable wireless networks. *IEEE Communications Magazine*, 52(2):196–203, 2014.

[25] K. M. Tsui, D.-J. Kim, A. Behal, D. Kontak, and H. A. Yanco. âĂIJi want thatâĂİ: Human-in-the-loop control of a wheelchair-mounted robotic arm. *Applied Bionics and Biomechanics*, 8(1):127–147, 2011.