Chapter 4

Building Intelligent Infrastructures

Steps toward Designing IoT-Enabled Library Facilities

Jonathan Bradley, Patrick Tomlin, and Brian Mathews*

"Only connect."

—E. M. Forester, 1910

Silently conversing objects surround us. From smartphones to Fitbits, invisible streams of data are coursing through and between the devices we hold in our hands or wear on our bodies. Our refrigerators, sensing their contents, churn out shopping lists or place orders on the web to replenish their stock; coffeemakers and lightbulbs now connect to Bluetooth and Wi-Fi networks. On a broader scale, the connectivity enabled by the Internet of Things (IoT) has been used to build "smart cities" with improved urban infrastructures and energy-efficient buildings. Sensors, beacons, accelerometers, and actuators: these and other components are the building blocks by which we increasingly digitize, organize, and personalize the physical world. As Jacob Morgan wrote for Forbes in 2014, "The new rule for the future is going to be, 'Anything that can be connected, will be connected.'"[1] The future is here.

Consider what IoT technology means for libraries. The traditional view of libraries as islands of automation, specialized expertise, and control over access to content holds less weight in a hyperconnected world. Yet libraries remain spaces immersed in data and data collection, a fact that IoT technology has the capability to harness in new ways. Imagine a library dashboard that not only tracks gate counts and usage of physical and digital collections, but also monitors the "health" and "fitness" of the building, from the cleanliness of bathrooms to the movement of furniture in areas of the library most heavily used for study or collaboration. Or imagine walking into a library commons and receiving recommendations on your phone about locations to sit based on the similarity of the research others are conducting nearby. Imagine a whiteboard that is able to push scholarly article recommendations based on the words, phrases, or diagrams written on

* **Jonathan Bradley** is the Innovative Technologies Coordinator for Learning Environments at University Libraries, Virginia Tech, where he evaluates and implements new technologies into library services. Projects that he has been a part of include building spaces for 3-D printing, virtual/augmented reality, and data visualization in the library. He is currently producing the hardware and software for the Smart Commons project, which gathers data about how patrons use library spaces by utilizing IoT technologies. **Patrick Tomlin** is the Director of Learning Environments at University Libraries, Virginia Tech. **Brian Mathews** is Associate Dean for Learning at University Libraries, Virginia Tech. He has previously serviced as an assistant university librarian at UC Santa Barbara. Brian frequently presents and writes on topics related to innovative technologies and organizational culture.

its surface. Enabling these connections—connections between people, and between people and devices—through IoT technology can empower librarians to make strategic decisions about library spaces and services and provide library users with a unique, personalized experience.

The migration to an IoT-enhanced library is a journey of multiple steps, of course. Some of these steps are infrastructural in nature, while others will require focusing on service design and the creation and delivery of a fluid user experience. Still others will entail the development of a system of software and algorithms to collect and aggregate library data in order to analyze it across space and time. This chapter provides a brief overview of our initial steps taken in such a direction. Over the past year, we applied IoT technology to the Newman Library at Virginia Tech in an attempt to better understand our users' interactions with its spaces. By tracking the movement of furniture in the library commons, we hoped to illuminate patterns of student work, examine the density of particular work areas, and ultimately create more effectively designed learning spaces and user experiences. Using accelerometers, motion detectors, force sensors, and Bluetooth beacons, we created a system for monitoring where and when furniture and equipment were moved, what study rooms were occupied, and how students interacted with them.

The project outlined here represents the first, preliminary steps in a much larger endeavor. Nevertheless, we believe it poses important questions for the study of library spaces and services at the outset. What metrics should frame the implementation of IoT devices? How do we get not only more data, but *better* data from the library itself? Can we effectively monitor the health of a building in terms of its physical condition? Is it possible to measure and articulate the fitness of our spaces in relation to the activities transpiring therein—that is, can IoT technology provide us with a more robust picture of the difference between the intentions for our spaces and how they are (or are not) actually used by library patrons? In short, can IoT technologies help us to better understand the nature of the interactions occurring in libraries and ultimately empower us to enhance the user experience in previously unknown ways?

We entered this experiment with an exploratory mind-set. Our purpose was both practical (What are the range of sensors available and how could we deploy them effectively?) and perspective-building (What types of data could we collect and what could it reveal about patterns in our learning environment?). Through this project we uncovered three overarching design challenges: battery life, programming language, and security. This section outlines the problems and offers some lessons learned.

Design Challenge 1: Battery Life

The most prominent design challenge while building the prototype for the Smart Commons module was battery life. According to the goals of the project, we wished to deploy numerous modules to chairs around our learning commons, meaning that maintenance would inevitably be a time-consuming job, and the battery life of modules could exponentially increase that maintenance time.

The original prototype had a battery life of just under one week, meaning ten modules would have to be located on the floor, removed from the chairs, opened, disassembled, charged, reassembled, and redeployed once again every week. Since the goal of the project was to eventually scale up and add more modules not only to our commons but to other branch libraries on campus as well, this model would not result in success. We determined that for it to be sustainable, battery life for a module would need to be closer to three months.

Lessons Learned

Anyone undertaking an IoT project in an academic setting needs to devote a great deal of thought to the hardware platform they will use. Boards like the Raspberry Pi, CHIP, and others in the family utilizing ARM processors seem like a good choice. They have huge communities of support, are easy to develop on, are cheap with lots of desirable features (built-in Wi-Fi, Bluetooth, etc.), can use multiple different programming languages, and are generally easy to obtain and get code running on. But they are poor choices for IoT projects because of the battery life of the device. As full computers, these boards draw power of a magnitude far greater than most embedded intelligent devices and simply aren't sustainable for a project that will not have wired power or need to be online for more than a few hours at a time.

At the same time, even when deciding to use a common IoT chip like an ESP8266, the type of board used merits examination. Development boards, like those in Adafruit's Feather series, are great for getting your project functional, but they may not be the best for the actual deployment. Many of these boards include features like onboard LEDs and USB serial bridges that help with development but that can hurt battery performance. Many onboard LEDs can be difficult to completely disable, and LEDs are a huge battery drain, even if running only while the chip boots from sleep. Other features can draw latent power even when not actively used.

For the second version of the Smart Commons module, we have switched from using a CHIP board to an ESP32 chip. Development is happening on an Adafruit Feather board, with the goal of having a

custom PCB cut for the project that contains only the minimum features needed to stretch battery life out as far as possible.

Design Challenge 2: Programming Language

Developing for IoT offers a plethora of programming language choices, and the decision about what to use for an academic project is not always clear-cut. For the Smart Commons project, we went initially with JavaScript since it is far easier to find a programmer in the library for JavaScript than for a language like C or even Python. The hope was that using JavaScript would make the project more accessible to interested developers at other institutions. JavaScript also offers in Node.js packages an abundance of tools for IoT development, and the Smart Commons project relied heavily on the Johnny-Five robotics and Bleacon packages for interfacing with sensors and hardware.

However, Node.js requires a full computer to run, which was fine for the ARM boards like Raspberry Pi and CHIP but does not translate to the small embedded chips like the ESP32. Programs using these lower-power chips are often coded in C or C++, which is a more difficult language to learn and offers more barriers to entry for an academic IoT project, in that the library has to have a C/C++ programmer to work on the project, which might not be a resource it has access to.

There are alternatives to using C/C++ on these IoT chips. By installing a different firmware, developers can enable a different language for development. Both the MongooseOS and Espruino firmwares support coding on IoT chips in JavaScript, and Micropython allows for Python coding on IoT platforms. However, these firmwares are just wrappers around the lower-level languages like C/C++, meaning that they are by their nature reactive to the underlying SDKs they are abstracting. This means that cutting-edge features often take a long time to gain support in these firmwares because they have to wait for the underlying SDK to code and stabilize a new feature, then the firmware's dev team (who are usually volunteers) have to code and test all of the wrappers before implementing.

Lessons Learned

The choice of programming language is a give and take in most situations, and the future of our Smart Commons project has been guided by this situation. The project has shifted from using JavaScript in the prototype module to using C++ as part of the Arduino IDE for the ESP32 that is at the core of version 2.0 of the project. This decision is the result of our need for cutting-edge features, namely BLE support for our chip. As of the writing of this paper, the only environment to have stable support for the BLE features of the ESP32 is the Espressif Systems (manufacturer of the ESP series of chips) SDK. However, since one of the goals is to make the code and the project as a whole as accessible as possible to other academic institutes, we will be monitoring other firmwares. We will likely be migrating the codebase to JavaScript or Python (or both) as the features we require become stable in those environments since JavaScript programmers are common and Python is currently the fastest growing programming language and easier to learn than C++.

For other groups pursuing similar IoT projects, we suggest using the most accessible language for your project that your technology needs allow. If you don't require cutting-edge features, Espruino or Micropython would likely easily meet your needs. However, if you do not plan to use open source and share your code with other groups and the project is intended for internal purposes only, then the choice boils down to the preferences of your internal development team.

Design Challenge 3: Security

IoT devices have received a great deal of attention recently because of security issues, which underscores the importance of taking extra steps to secure projects before they end up as part of a botnet that damages the health of the internet as a whole. For the Smart Commons project, we were warned by our central IT service that our campus has a large number of hackers attempting to gain access to the university's secure systems. Our IoT devices can't provide them that kind of access, but that doesn't mean that they can't be used as part of an attack or compromised to serve some other nefarious purpose. This meant we would need to make sure we took the security seriously from the outset.

In addition to software and networking security, physical hardware security was also a concern, meriting forethought to accomplish the project in a way that doesn't leave an IoT project completely open to attack. Our modules are in public places within the reach of patrons, meaning issues like theft and direct tampering also had to be considered when designing the modules. These are challenges that can be overcome, but it is our responsibility, as the stewards of data collected from our students and patrons, that we not be reactionary to attacks but instead proactive to mitigate as many risks ahead of time as possible.

Lessons Learned

In addition to the standard security practices, like changing all of the default login passwords to long,

random strings, locking down unused ports, and securing API endpoints, there are a number of other practices that we put into place with the Smart Commons to head off attackers. First, we ensured that our data reporting used encryption for security. SSL certs are now free and easy to obtain, so there is no real excuse for transmitting data over insecure connections. It takes more work in the code, but it should be the default for an IoT project, even if you don't think the data you'll be transmitting is sensitive.

Second, we made the decision to have our module not act as a webserver. We knew that in addition to reporting data, the module would need to receive some information as well, mostly about alarm states and internal matters. Originally, the plan was to have each device act as a webserver and listen to information only for specific sources. However, webservers are targets for hackers and sources can be spoofed, so if at all possible, avoid having your device act as a server. We realized that with the data we needed to receive, we could just have the device check a state against our known data source during other operations instead of always listening and responding to incoming HTTP requests. While some IoT projects will inevitably require the device to act as a webserver, as you begin the project, you should consider whether or not your device could receive the needed information in some other way, like polling a trusted source periodically or grabbing the data during other operations.

In addition to software concerns, it is also important to think about physical access to the modules. It became clear to us during our design process that it would be easy for a patron to simply steal one of our modules from underneath a chair without some sort of physical security or to attempt to hack the device via physical connection to one of the ports on the chip. To mitigate these problems, we worked an alarm system into the design with button triggers that would set it off. We also designed a custom 3-D printed case for the module that would trigger the alarm if opened or if removed from the chair and would restrict access to things like ports and pins. With a bit of clever thinking, it is also possible to hide the screws that remove the case from the chair behind the case itself, making it so that one would have to open the case, thus setting off the alarm already, just to get access to the means for removing it from the chair completely. The alarm sounds only a rather quiet buzzer, just enough to let patrons know that they have done something wrong without disrupting an entire floor of students studying. More importantly, the alarm system also sends an email to the team informing them of the tampering and providing the device's last known location (gathered thanks to the BLE Beacon location monitoring). Additionally, it is advisable to purchase a board with encryptable flash space so that if someone does manage to run off with a device, that person will be unable to get access to the code and the API access or other sensitive information contained therein.

Metrics and Sensors

Due to the inexpensive nature of sensors and the wide variety already available for purchase, the metrics that can be gathered with IoT devices are nearly endless. The first prototype of the Smart Commons module tracked location-based data via Bluetooth, movement data through an accelerometer, and force data through a force-sensitive resistor. For the second iteration, the accelerometer was dropped from the design because the data gathered was deemed less useful than the location-based data being returned via the Bluetooth interface, and removing it increased battery life while reducing both cost and size. The second iteration of the Smart Commons was focused on refining the architecture, the battery life, and the size of the module, so no new sensors were added. As this module matures, the third iteration will add new functionality that we have identified as being desirable for assessing the health and fitness of the building.

Other sensors and metrics we intend to implement in the future include water leak detectors and sensors that can register humidity, temperature, air particulates, and barometric pressure. Each of these will assist in delivering information about the health of the building. From inoperable air conditioners to burst pipes, time-sensitive facilities information can be relayed to a dashboard immediately; adding additional sensors to the existing deployment of Smart Commons modules around the building is cheap and easy with tangible benefits for user experience and service design.

Additionally, we have been planning companion modules for the main Smart Commons module that can provide additional data to augment what is already gathered. These companion modules will utilize door open/close sensors, PIR (passive infrared) sensors, Velostat pressure-sensitive sheets, and thermal cameras to better track the fitness of the building. With these sensors we can better understand through anonymous data where students are at in the spaces, whether they are working together or separately, and the frequency with which they migrate to other places in the building for different task-based learning. The eventual goal would also be that these sensors could provide information on how students are using our services: Are they coming to the building to specifically use a service like one of the library's technology-oriented studios and staying to study, or do they use these services because they are already in the building doing other things?

These companion modules will likely require a different board architecture and power scheme than the

main Smart Commons modules, but they will report back to the same data-gathering and dashboarding system, allowing them to augment the snapshots we get of the building's health and fitness. And with the low cost of these sensors, it is easy and cost-effective to add new ones as a new need for data gathering becomes clear. Most of these basic sensors are available for less than five dollars, and even most breakout boards carrying more complex sensors and interfacing can be had for less than fifteen dollars. By far the most expensive portion of the current Smart Commons project is the thermal camera purchased as a test prototype for tracking patrons in a space, at just over $200. A normal camera (costing around twenty-five dollars) could have been used for this, but a thermal camera was purchased to meet Virginia Tech's IT privacy guidelines for cameras in public spaces, which dictate that we should avoid having student faces captured on network-connected cameras.

IoT is still a rather fledgling technology despite the fact that many of these sensors have existed in some form or another for decades. As demand for them increases, the cost of sensors will continue to drop, and new sensors will be developed to meet emerging needs. This means the potential for data collection is huge; the question for librarians becomes less about what information could be gathered and more about the creation of purposeful, well-defined metrics and assessment strategies.

Conclusion

From checkout statistics to website analytics, libraries have long invested in data collection as a means of creating, measuring, and improving services. As more libraries have focused on assessing user experience and gauging the impact of their spaces, greater prominence has been given to user studies employing ethnographic strategies such as observations and interviews. What these approaches lack, however, are both the real-time results offered by IoT technology and the broader picture of the library it provides.

Library buildings are evolving. Now they can do much more than provide passive spaces for people to learn and work. It is when sensor-based applications and objects are aggregated to form a choreographable system that they have the potential to transform the library. True smart buildings are more than the sum of their IoT technologies—they utilize an intelligent infrastructure driven by an integrated network of systems and analytics. Similarly, building an intelligent infrastructure for libraries requires seeing them holistically, less like a container and more like a living organism in a state of constant flux and flow.

The University Libraries at Virginia Tech have started on this path. Each iteration brings us closer to realizing the potential of these sensor technologies. Since the IoT is still in an early stage, we are using each step to determine feasibility and the range of possibilities. Our goal is not only to better understand the health and fitness of our facilities and to ultimately improve services for our community, but we also aim to inspire other libraries to explore IoT and connect their buildings with ours.

Note

1. Jacob Morgan, "A Simple Explanation of 'The Internet of Things,'" Leadership/#NewTech, Forbes website, May 13, 2014, https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#30029f7d1d09.