IST-Africa 2016 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2016
ISBN: 978-1-905824-54-0

A Design Approach to Adapting Maker Community Projects to the IoT Constrained Device Philosophy

Andrew Cyrus SMITH
CSIR Meraka Institute, PO Box 395, Pretoria, 0001, South Africa
University of South Africa, PO Box 392, UNISA, 0003, South Africa
Tel: +27 12 8414626, Fax: + 27 12 8414720, Email: acsmith@csir.co.za

Abstract: Internet of Things (IoT) advocates promise huge benefits but what technical challenges does the maker community face in order to participate in this new technological wave? We report on our experience in incorporating a maker community friendly weather station with an IoT system and in so doing identified the low energy design philosophy as a challenge not previously significant in Internet-connected systems. This challenge highlights some of the restrictions that the IoT design philosophy poses on the maker community. Our research approach was well suited for this challenge. We describe the Design Science Research approach followed, the challenge that the resource constrained IoT system posed, and our solution to this challenge.

Keywords: Internet of Things, IoT, maker community, resource constrained.

1. Introduction

We address the problem of transiting from generic Internet solutions to a solution for a specific Internet of Things (IoT) system.

The anticipated value of the IoT lies in its pervasiveness to the point where every single light bulb [8] (for example) is an endpoint in this network and individually addressable. This vision is only feasible if the additional cost and energy consumption of the billions of “smart” devices at the IoT periphery are small. This requirement dictates that any endpoint electronic circuitry will be memory constrained and have little processing abilities. Therefore, in order for the IoT to succeed both the IoT architects and engineers have to adopt a constrained device design philosophy.

Usually, integrating a weather station with a data logging and dissemination system does not pose the challenges we faced in this research project. This is because, in general, the capability of computing and dissemination systems have kept pace with component improvements as these approached Moore’s [13] prediction. However, the supply and storage of electrical energy has not. The result is that while the ability to generate and process data has increased significantly, the supporting energy provision has lacked and devices remain constrained by their ability to store energy. A well-known case in point is the modern smart phone that has to be connected to a power source daily whereas the previous generation of phones (feature phones) would last for days on a single charge.

IoT architects are acutely aware of the energy dilemma and have designed communication protocols to address this challenge. Another approach towards reducing the energy requirement is to increase the time interval at which subsystems exchange data. As our research progressed, we discovered that the challenge in migrating well-established design principles to IoT based solutions was not only the adoption of the CoAP [11] protocol but (and more useful to us) was an increased communication interval.

Amongst other sensors and actuators, we have identified a maker community friendly weather station with which to evaluate our IoT platform. The maker community is an extension of the do-it-yourself (DIY) culture [15]. Whereas DIY is mostly the activity of a private individual, the maker movement emerged due to the low barrier of entry to online collaboration and information resources facilitated by the Internet [1, 4, 5, 14]. Privately owned and erected weather stations are popular maker community projects. One reason for their popularity is that they have the potential to make geographically fine-grained weather data available to weather researchers. In turn, these researchers are afforded additional data with which to improve their understanding of the changing weather patterns caused by global warming. In addition, the local communities that surround these amateur weather stations have access to localised weather conditions in close to real time. Based on this, we deemed it worthwhile to incorporate a maker community friendly weather station with our IoT system.

Numerous weather stations are commercially available to the amateur weather forecaster from which to choose. Many amateurs are also members of the maker community. However, most of these commercial stations have inaccessible proprietary interfaces through which data are published to the Internet and from where the public can retrieve the results. These stations are often also expensive and therefore not attractive to the maker community.

The combination of propriety interfaces and high cost contrasts with both our open source approach to research and our understanding of the maker community as characterised by a DIY attitude using affordable resources. Our solution was to combine low cost mechanical sensors with low cost electronic circuitry and develop a maker community friendly open design weather station.

At the onset of our project, we were aware of the need to develop a maker community friendly interface between three weather sensors and an existing IoT system. Our initial solution was to incorporate an Arduino to sample the fast changing sensor signals and produce a slower and varying signal suitable for the ActivePlug front-end of the IoT system. Our first implementation using the Arduino served this function.

After evaluating the solution, we observed that the 0.5 second sampling intervals were suitably catered for but as the sensed data propagated up the IoT stack to the Gateway we noted that data were missed due to the 5 second Gateway update interval. This prompted the need for a second iteration that would cater for the low update rate. Evaluation of the second iteration indicated that we had achieved our goal and we subsequently terminated the project at this point. The result is that a new set of operational principles have been added to our knowledge of how to integrate maker community friendly weather sensors with an IoT system.

In section 2, we describe our research objective by comparing a "traditional" Internet-based system with an IoT solution. Section 3 provides an overview of the Design Science Research methodology that we followed in our inquiry. The technology applied in this research is introduced in Section 4, specifically an overview of the IoT system and the sensing system that took form as weather station sensors. Section 5 reflects our results. We conclude with Section 6 and suggest future research.

2. Objective

The IoT is receiving a significant amount of publicity, both in the research community and with the general public. Soon, the maker community will enter the promising world of the IoT, if not already. Our objective is to determine what design approach changes should be considered when migrating maker projects from "tradition" Internet based systems to IoT systems.

Figure 1 illustrates the objective with the top section reflecting a conceptual design outline of typical well-known Internet-based solutions. The bottom highlights the unknown element that is also the question posed here. Our research question is: "what design thinking change is required when familiar Internet-based sensing systems are migrated to IoT solutions?"

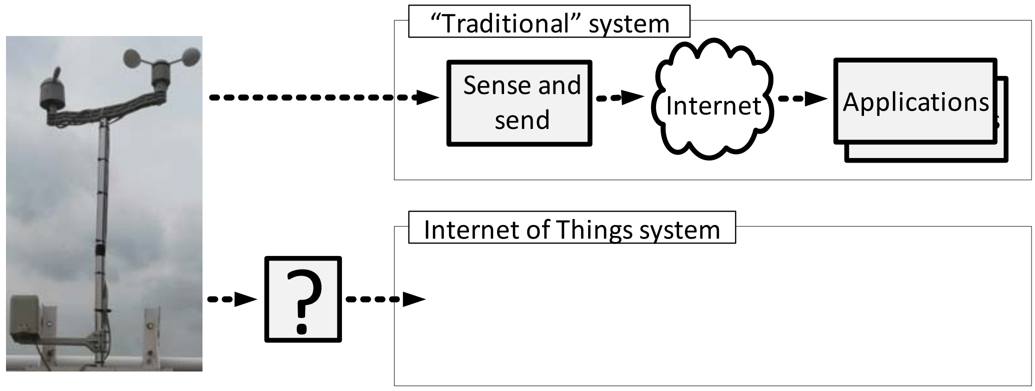

Figure 1: Objective summary

3. Methodology

Interfacing sensors such as those of a hobbyist weather station to the Internet is not usually considered research. The reason for this view is that hobbyist weather stations have matured to the point of being "plug-and-play"; that is, once the hobbyist has determined a spending budget it is a simple task to find a supplier, purchase and install the station, and connect it to the Internet. What makes the current project interesting is that no hobbyist weather station exists that can connect to the IoT. However, the maker community often leads the adoption curve along with researchers and it is this community that we hope will find our research results useful.

It is in our design approach that we differentiate our research from the weather enthusiast's activities. An enthusiast relies on tested and established design principles when installing a weather station. This approach is similar to that of an engineer. For both, the goal is to minimise risk [18] and any design activity they conduct is known as "routine design" [10]. None-the-less, the engineer, the researcher, and the weather enthusiast are all called designers [17]. What differentiates the researcher from the routine designer is the goal to create new knowledge whereas the goal of the engineer and the enthusiast is to create new experiences [3].

A research methodology that is well suited to the current project is Design Science Research and the first publication on this research was by Hevner [9, 10]. This methodology is well suited when artefacts are created and applied to solving organisational problems [10]. The methodology assumes that not all the required knowledge is available at research onset and that this knowledge will be added to as research continues. This is achieved through iterative and incremental research cycles that consist of an awareness of the problem, a suggested solution, the development and evaluation of the suggested solution, and a conclusion when the research terminates. According to this methodology, it is acceptable to exit the current iteration at any of these process steps.

4. Technology Description

The research project consists of various subsystems (Figure 2**)** that include electro-mechanical sensors, electronic hardware, embedded firmware, and server software. The focus of this paper is the challenge that the maker community faces when the mechanical sensors and electronic circuitry have to interact with the server software hosted on the Internet. Conventionally, this interaction is via a gateway and the design of such a gateway does not put the conservation of electrical energy high on the list of priorities. In contrast, emerging IoT systems make energy conservation a priority. Consequently, designing systems that conserve electrical energy are becoming relevant to the maker community.

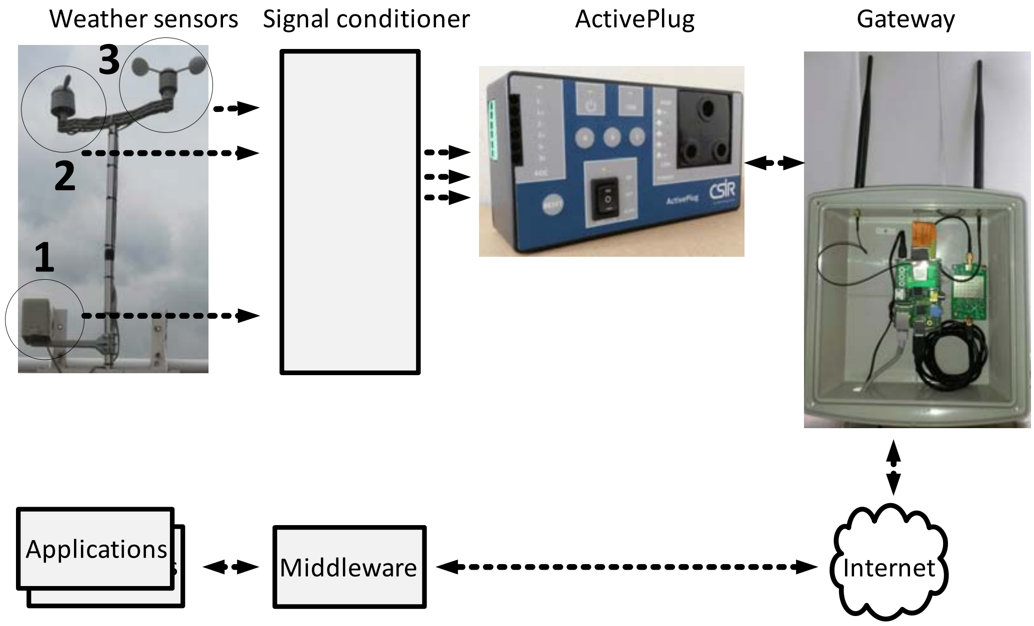

Figure 2: System overview

4.1 The Internet of Things system

Our IoT system design is hierarchical with three identifiable layers. These are the sense and actuation level, the middleware, and the application layer. The sense and actuation layer is the focus of this paper and it is at this level where the IoT's reliance on constrained devices has the biggest impact on the maker community. To clarify, consider one of the selling points that advocates of the IoT widely share, that is, lowering the electrical power consumption of devices at the sensing and actuation layer.

With the envisioned deployment of billions of IoT-enabled devices globally, energy conservation has to be considered as a major design criterion. Standards have therefore been developed with the aim to reduce the energy requirements of IoT systems. In the case of wireless communication another criterion is bandwidth usage. This is because the electromagnetic spectrum is a limited resource and every data bit transmitted adds to the energy demand. A relevant standard is the CoAP communication protocol that serves a similar purpose as what the UDP/IP protocol suite does for the Internet. Historically, CoAP was custom developed for the IoT with reduced packet data fields and also collapsed data layers. Conserving energy was one of the driving factors in its development. By reducing the size of the packets and limiting the number of layers, it is argued that less energy is required when communicating.

Although no standard yet exists for the design of an IoT system, we based our design on existing and published IoT research results, the anticipated trajectory of IoT development, and information systems best practice. The result is a multitier system [12] that connect the real-world actuators and sensors with abstract decision making systems. This system has already been applied in the TRESCIMO joint European Union/South African research project [6].

Two identifiable components connect these layers and we call these the Middleware and the Gateway. The Gateway acts as a channel through which the sensors and actuators communicate with the Middleware. The Middleware in turn keeps track of the exchanged data and maintains a digital representation of the physical world. Decision-making applications retrieve data hosted on the Middleware and sends commands to the physical world using the same channel.

The Gateway is core to the current discussion. Not only need the communication protocol of the gateway and the sensors/actuators be compatible with each other but the communication bandwidth between these levels have to be matched.

Mismatched communication bandwidth requirements are a challenge for the maker community. To elaborate, the maker community has up to now either interfaced their devices to stand-alone systems or to Internet-based systems. Both systems are well established and design practices have been refined. The IoT low energy design philosophy introduces a new dimension to the design process of sensors and actuator sub-systems.

4.2 The weather station

The maker community friendly weather station consists of a commercial sensor set and a low cost Arduino [2] circuit. Sensors measure wind speed, direction, and rainfall. An anemometer turns in the wind and an electrical pulse is generated for each rotation. Rainfall is detected in a similar way: Two small water buckets are mechanically joined on a tipping arm, with only one bucket a time in position to collect fall rain. When the bucket is full, gravity makes it tip and empty. At the same time, the arm lifts the connected bucket into position to receive the rain. Every tipping results in a short electrical pulse.

The wind direction measurement is more complicated and uses a set of switches connected to resistors of unique values. Unfortunately, these values are not according to a linear progression and therefore proper interpretation of the signal necessitates the use of a lookup table. The purpose of the non-linear progression remains unclear but indications are that it is a variation of Grey code [16] to assist with error detection when the sensors fail.

The challenges discussed up to this point are not unique to IoT systems but are also faced by all maker enthusiasts that sense weather conditions, irrespective of what the weather sensors are connected to. It is the low-power IoT philosophy that adds an additional challenge and specifically the rate at which data is sampled.

In order to conserve energy, both the ActivePlug and the Gateway have sampling intervals that are measured in seconds. This rate is in sharp contrast to “traditional” systems that measure real-time events such as those in a maker community friendly weather station. The ActivePlug samples its analogue inputs twice a second whereas the Gateway updates data only every five seconds. It is clear that these rates are incompatible with the rate at which data are generated.

Returning to the rain and wind speed sensors: one switch in each sensor produces very short pulse for each bucket tip and for each revolution, respectively. The pulse is very short and requires a microprocessor sensing mechanism that is more sophisticated than polling. A solution is to use a microprocessor interrupt routine that can intercept the short interrupts and process the signal into something compatible with the IoT system. For this function, we developed an Arduino Pro Mini-based subsystem. Using this, the short pulses received from the two respective sensors are converted into output signals that have a much longer duration. This is sufficiently long to be sensed by the ActivePlug’s polled input and therefore overcome the mismatch between the rate at which data are generated and the rate at which data are processed. The Arduino subsystem therefore not only senses the very short pulses coming from the sensors but it also processes these signals so that they are compatible with the IoT system.

In the above, we have addressed the incompatibility between the fast varying sensor data and the ActivePlug's long sampling interval. The next challenge was to reduce the rate at which our generated signal changes so that it would be compatible with the Gateway's 5 second communication interval. We made a design decision not to alter the firmware of either the ActiveGate or the Gateway but instead use the Arduino for this purpose. Our solution is to count the number of pulses received in a 5-second interval and generate an analogue voltage on an Arduino pin. The voltage thus generated is in direct proportion to the number of pulses counted in the 5-second interval. This voltage is maintained for the duration of the interval, resulting in a signal that does not exceed the rate at which the Gateway sends data to the remainder of the IoT system.

However, the Arduino does not produce true analogue voltages. Instead, the "analogue" output is a pulse width modulated binary signal varying between 0V and 5V. In order to get a better analogue signal, we applied a first order low-pass passive RC filter on this pin. A cut-off frequency was chosen according to Nyquist's [7] theorem and to suit the 5 seconds transmission period of the Gateway. We calculated the cut-off frequency by applying Nyquist's theorem that relates signal sampling to bandwidth. A single pole passive RC circuit was therefore implemented and added in series with the signal at each of the analogue output pins on the Arduino. In effect, the processing algorithms combined with the low-pass filters remove all fluctuations that are shorter than the 500 millisecond and the 5-second sampling intervals.

5. Results

Designing for the IoT leads to complicated systems because these have to make provision for an almost infinite number of data producers and data consumers. This is in sharp contrast to traditionally engineered systems that cater for no more than dozens of data sources. Our IoT system addresses this complexity to some extent by the digital model of the physical world it constructs and maintains. The design principle is for a peripheral to add itself effortlessly to the digital model. Ultimately, applications will query the model and subscribe to relevant data sources. Although we have a working mechanism by which a device is automatically removed from the model, we have yet to implement the same for new peripherals.

Fault finding is a recurring problem in this and other IoT systems we implemented. The reason is that once data has entered the IoT system it remains a highly specialised problem to determine why the system does not behave as expected. Finding the root of the problem requires the assistance of the IoT system designers and coders.

Using available resources we determined that assumptions related to the flow of data between subsystems have to be revisited. For example, sensors data on its way to a user application might reach the first IoT software component but never reach the last component where it will be rendered for the user. Well-established tools such as the Unix ping and traceroute commands are useful to find faults at the lower layers in the IoT protocol stack, but what is now required is a simple tool to help determine why data do not propagate all the way to the user application.

The result is a workable system that will likely suit the needs of the maker community, albeit one of reduced resolution and accuracy when compared to existing non-IoT solutions.

6. Conclusions

In this paper, we explained a new problem faced by the maker community when designing sensing systems that will interact with IoT systems. This challenge is due to the low energy philosophical approach adopted by IoT designs. Such a stance supports widely deployed systems that can operate from battery power for extended periods. However, this stance now necessitates a redesign of conventional sensing circuits that have to date not been limited by low power design philosophies.

Using as example the three sensors of a weather station and an IoT system, we detailed the need for adapting current circuit design thinking and elaborated on an adapted design approach. By means of the new design approach, we successfully interfaced rudimentary sensors with an IoT system. However, in order to accomplish this we had to discard information contained in the original signals. In addition, the resultant data are mere indicators of the original values and we have yet to design a mechanism with which the processed data can be calibrated against the actual inputs.

In our endeavour to improve our designs, we hope to expand our fault finding tools. Such tools are valuable in both the research phases of our projects and during deployment, including customer support. A tool we envisage is a data flow dashboard that will help make the invisible world of dataflow accessible in real time, at least as a way to observe the flow of data between the multitude of architectural hardware and software layers. Access to various levels are needed, starting at the highest and most abstract layer. In a deployment, debugging will typically start at this layer and progress to the lower levels. Eventually, the specific device can be identified where the data flow is blocked. This would however not be the end of the investigation because the device will have its own layers consisting of combined hardware and software elements.

References

[1] Anderson, C. 2012. *Makers: The new industrial revolution*. Crown Business.

[2] Blum, J. 2013. *Exploring Arduino : tools and techniques for engineering wizardry*. John Wiley & Sons, Inc.

[3] Bonsiepe, G. 2007. Design research now : essays and selected projects. R. Michel, ed. Birkhäuser. 25–40.

[4] Calderon, R., Fels, S., Anacleto, J., Memarovic, N. and Thompson, W.T. 2014. Hacking HCI3P: Second Workshop on Human Computer Interaction in Third Places. *Proceedings of the 2014 Companion Publication on Designing Interactive Systems* (Vancouver, BC, Canada, 2014), 195–198.

[5] Cermak-Sassenrath, D. and Møllenbach, E. 2014. Teaching to Tinker: Making As an Educational Strategy. *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational* (Helsinki, Finland, 2014), 789–792.

[6] Coetzee, L. et al. 2015. TRESCIMO: European Union and South African Smart City contextual dimensions. (2015).

[7] Diniz, P.S., Da Silva, E.A. and Netto, S.L. 2010. *Digital signal processing: system analysis and design*. Cambridge University Press.

[8] Gershenfeld, N., Krikorian, R. and Cohen, D. 2004. The Internet of Things. *Scientific American*. 291, 4 (2004), 76–81.

[9] Hevner, A. and Chatterjee, S. 2010. Design research in information systems: theory and practice. Springer. 9–22.

[10] Hevner, A.R., March, S.T., Park, J. and Ram, S. 2004. Design science in information systems research. *MIS Quarterly*. 28, 1 (2004), 75–105.

[11] Kovatsch, M. 2013. CoAP for the Web of Things: From Tiny Resource-constrained Devices to the Web Browser. *4th International Workshop on the Web of Things (WoT 2013)* (2013).

[12] Madhoo, H., Khatri, A., Willemse, T., Oosthuizen, D. and Coetzee, L. 2015. Future Internet concepts for demand management. *Domestic Use of Energy (DUE), 2015 International Conference on the* (Mar. 2015), 19–26.

[13] Meindl, J.D. 2003. Beyond Moore's law: the interconnect era. *Computing in Science & Engineering*. 5, 1 (2003), 20–24.

[14] Mohomed, I. and Dutta, P. 2015. THE Age of DIY and Dawn of the Maker Movement. *GetMobile: Mobile Comp. and Comm.* 18, 4 (Jan. 2015), 41–43.

[15] Oehlberg, L., Willett, W. and Mackay, W.E. 2015. Patterns of Physical Design Remixing in Online Maker Communities. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea, 2015), 639–648.

[16] Savage, C. 1997. A Survey of Combinatorial Gray Codes. *SIAM Review*. 39, 4 (1997), 605–629.

[17] Simon, H.A. 1996. *The sciences of the artificial*. MIT Press.

[18] Vaishnavi, V. and Kuechler, W. eds. 2013. Design Science research in information systems.

