Using the Internet of Things to Enable Electrical Load Optimisation

Laurie BUTGEREIT, Andrew SMITH
Meraka Institute, CSIR, Pretoria South Africa

Abstract: The Internet of Things is a new paradigm of more and more physical objects, or "things", becoming connected to the Internet. As more and more physical objects are enhanced with digital intelligence, the Internet of Things grows. The communication with the objects could be uni-directional. In such cases, the object merely posts a status of itself or updates information about itself on a server. More interesting, however, is when there is bi-directional communication allowing the object to post information about itself and also allowing the object to be controlled by some process. This paper describes a research project in balancing the electrical load in the kitchens of an IT organisation. Ten kitchen appliances were enhanced with digital intelligence and could send back information about electrical consumption. In addition, however, the appliances could be switched on and off depending on certain conditions. This paper would be of interest to any readers who need to balance electrical load due to physical constraints or for financial constraints.

Keywords: iot, things, electricity, load balancing

Introduction

One of the fundamental problems with electrical energy is that it is very expensive to store electricity once it has been generated [1]. Electricity must be used as soon as it is generated. If it is not used and if it is necessary to conserve the energy, then the electricity must be converted into some other form of energy before storage. For example, electricity could be converted into chemical energy and stored in a battery. Or electricity could be converted into potential energy by pumping water up a hill and storing it behind a dam. One side effect of this is that it is difficult to store electricity during times of low demand for later consumption when demand for electricity is high.

Commercial electricity generating companies estimate future electrical consumption by society. Because electrical energy is so difficult to store, it is usually required that electricity companies have sufficient generation capacity in order to satisfy peak consumption needs. That is the ideal situation. Since late 2005, however, South Africa has been subject to a system of rolling blackouts (also called load shedding). These rolling blackouts reached crisis level in 2008 when the estimated short fall of electricity was 5000 MW [2].

Besides the problem of maximum electrical consumption, electrical supply companies often have different unit charges for electrical consumption depending on the maximum amount consumed.

This paper describes a project involving the use of Internet of Things technologies where electricity consumption is levelled or balanced by monitoring and controlling consumption via the Internet. This paper would be of interest to readers who need to limit the maximum consumption of electricity for either physical reasons or for financial reasons.

The Internet Of Things

The Internet of Things is the paradigm of an exponentially increasing number of physical objects being connected to the Internet. These physical objects are expected to become active participants in business processes, information processes and social processes where they are able to interact and communicate among themselves and with the environment [3].

The Internet of Things could be considered to be the next step beyond "ubiquitous computing" [4]. The Internet of Things describes a situation where an increasing number of physical objects have embedded intelligence. It is anticipated that these physical objects will communicate with the Internet by using a variety of communication mechanisms that include wireless radio, radio frequency identification (RFID), satellite, wired Internet, bar codes, QR codes, and fiducials [5].

Examples of these intelligent objects include luxury cars that are equipped with satellite tracking devices and pets with RFID tags medically embedded under the skin. Even a human heart has joined the Internet of Things [6].

But one of the more interesting challenges on the Internet of Things is intelligent decision making based on real-time readings from sensors [7]. Through the use of intelligent decision making, rapid responses can be given to physical phenomena.

Research Methodology

This project used a Design and Creation Research Methodology as defined by Oates [8]. The Design and Creation Research Methodology is a five step iterative methodology. The five steps are:

1. Awareness – the recognition of the problem and the statement thereof
2. Suggestions – potential ideas on how this problem could be solved
3. Development – implementation of these ideas
4. Evaluation - evaluation of the developments
5. Conclusion – concluding remarks

There were four major iterations of these steps. These four iterations (each containing a number of sub-iterations) were concerned with:

1. Monitoring the electrical consumption of the ten electrical appliances
2. Switching the electrical appliances on and off
3. Creating a model which indicated when specific appliances should be switched on or off depending on electricity consumption
4. Implementing that model

Each of these four major sections will be described in detail.

Beachcomber Framework

For this project, the Beachcomber framework was used. Beachcomber is a JEE (Java Enterprise Edition) Mobicents application which is designed to link the Internet of Things with the Internet of People [9]. The Beachcomber framework consists of three levels as can be seen in Illustration 1 (depicted with solid lines) and the physical layer is a fourth level (depicted with dashed lines). The fourth level connecting to the physical world can use any of the Resource Adaptors. The fact that the illustration only shows them using two specific Resource Adaptors is only for artistic purposes.

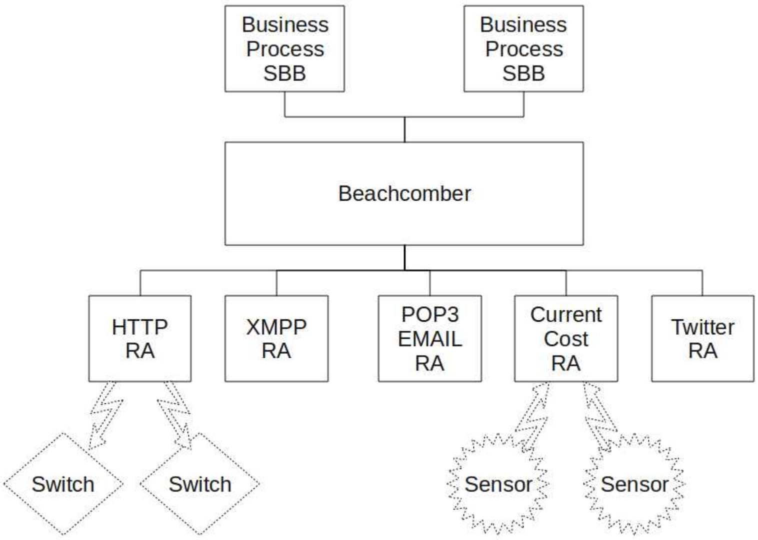


Illustration 1: Beachcomber framework with physical layer

The lowest level of the Beachcomber framework consists of the Resource Adaptors. In Illustration 1, the Resource Adaptors are labelled with the suffix RA. These Resource Adaptors communicate with the physical world using various protocols including HyperText Transfer Protocol (HTTP), eXtensible Messaging and Presence Protocol (XMPP), and Post Office Protocol version 3 (POP3) Email. Depending on the type of physical switch or sensor used, the appropriate Resource Adaptor is configured to read data from them or write data to them. In some cases, again depending on the type of sensor or physical switch, new a new Resource Adaptor must written as is the case with the CurrentCost RA.

When a Resource Adaptor receives a message, it converts the message to a common format and forwards it to the Beachcomber routing facility. The routing facility refers to configuration tables and forwards the information to a business process. The business processes are implemented as Mobicents Service Building Blocks (SBBs). These SBBs process the information and then send replies back to the routing facility which then forwards the replies to one or more Resource Adaptor.

Monitoring Electrical Consumption

The five steps in the design and creation research methodology were traversed in setting up a mechanism to monitor electricity consumption.

> 1.Awareness – the researchers were aware of the fact that the electricity consumption needed to be monitored.
> 2.Suggestions – it was suggested that Current Cost equipment be installed and appropriate software written to capture the consumption data.
> 3.Development – the equipment was installed and the software was written
> 4.Evaluation – the results were evaluated
> 5.Conclusion – it was concluded that the commercial Current Cost equipment would be satisfactory for this project.

The commercial Current Cost equipment (as can be seen in Illustration 2) is supplied with clamps which are attached around the cables that power the appliances that are monitored. No electrician is required to install the Current Cost equipment. The data from the clamp are communicated using a wireless device base station. In turn, the base station is equipped with a USB connection which can be accessed by a computer workstation.

Illustration 2: Current Cost equipment (Image from www.currentcost.co.uk)

The Current Cost base stations generates output in the eXtensible Markup Language (XML) format. A Resource Adaptor was written (indicated as the CurrentCost RA in Illustration 1) to parse the XML message from the Current Cost base station, and forward it to the Beachcomber routing facility.

At the time of writing this paper, ten appliances were connected in two different kitchens on two different floors of the building. Each kitchen had the same number and type of appliances. In each kitchen a geyser (hot water heater), kettle, microwave oven, refrigerator, and coffee machine were connected.

A sample of the XML sent by the base station to the CurrentCost Resource Adaptor can be found in Table 1. Of primary importance is the sensor number (which identifies the appliance) and the number of Watts being drawn by the appliance. When an appliance was off, the Watts value was close to zero. When an appliance was turned on, the Watts value indicated the number of Watts of electricity being drawn. There were some minor calibration problems and Watts values less than ten were considered as the appliance being in the off state.

This information was sent to the CurrentCost Resource Adaptor multiple times per second which enabled the Beachcomber platform to receive more or less real-time information.


```
<msg>
<src>CC128-v0.12</src>
<dsb>00149</dsb>
<time>13:58:10</time>
<tmpr>24.0</tmpr>
<sensor>5</sensor>
<id>03076</id>
<type>1</type>
<ch1>
    <watts>00005</watts>
</ch1>
</msg>
```


Table 1: Sample XML from Current Cost equipment

Creating The Load Balancing Model

After monitoring the information sent by the CurrentCost base stations, a pattern of consumption was observed. The geysers (also called hot water heaters in parts of the world) and the kettles consumed the most electricity. The geysers consumed approximately 1.5 kVA and the kettles consumed approximately 2 kVA. The other appliances consumed substantially less electricity.

The first two steps of the iterative process as defined by Oates [8] was traversed.

1. Awareness – the researchers were aware of the fact that model needed to be created that would balance the electrical load without inconveniencing employees
2. Suggestions – it was suggested that the geyser be turned off whenever any other appliance was switched on.

The remaining three steps in the iterative process were traversed in later phases of this project.

The model divided the appliances into two types of appliances. *Permanent* appliances were essentially on all the time and did not need to react to human demand. There were two appliances in the *permanent* classification: geyser and refrigerator. *On-demand* appliances were appliances which were manually switched on and off by humans and the humans were usually waiting close by while the appliance was on. There were three appliances in the *on-demand* classification: kettle, microwave, and coffee machine.

In general, the *permanent* appliances were switched on for longer periods of time than the *on-demand* appliances. In other words, when the geyser was switched on, it remained on for five minutes. The microwave, in contrast, was only switched on for approximately minute when in use.

The initial model we implemented followed these rules:

1. No two *permanent* appliances could be on at the same time
2. If an *on-demand* appliance was turned on, any *permanent* appliances were turned off.
3. After the *on-demand* appliance was turned off, the *permanent* appliance was turned on again.

It is important to note at this point, that our model applies to an office environment where employees are using the microwave merely to heat food up. In the case of home usage where the microwave is used for longer periods of time to actually cook raw food, our model may not be applicable.

Implementing the Model

The development step in Oates' design and creation research methodology was traversed during the implementation of this model.

A Mobicents Service Building Block was written which would receive information from the Beachcomber routing facility. It received timeous information (in the form of the sample XML message found in Table 1). From this information, the Service Building Block could keep track of the state of all ten appliances. As messages were received, the Service Building Block compared the indicated Watts consumed by the appliance to the Watts stored from the previous message. Taking into account minor calibration errors with Watts below ten, the Service Building Block could determine when an appliance was turned on within seconds of the event.

Prior to actually switching the geyser off and on, the Service Building Block merely posted on Twitter what was happening in the kitchens and what the model should be doing

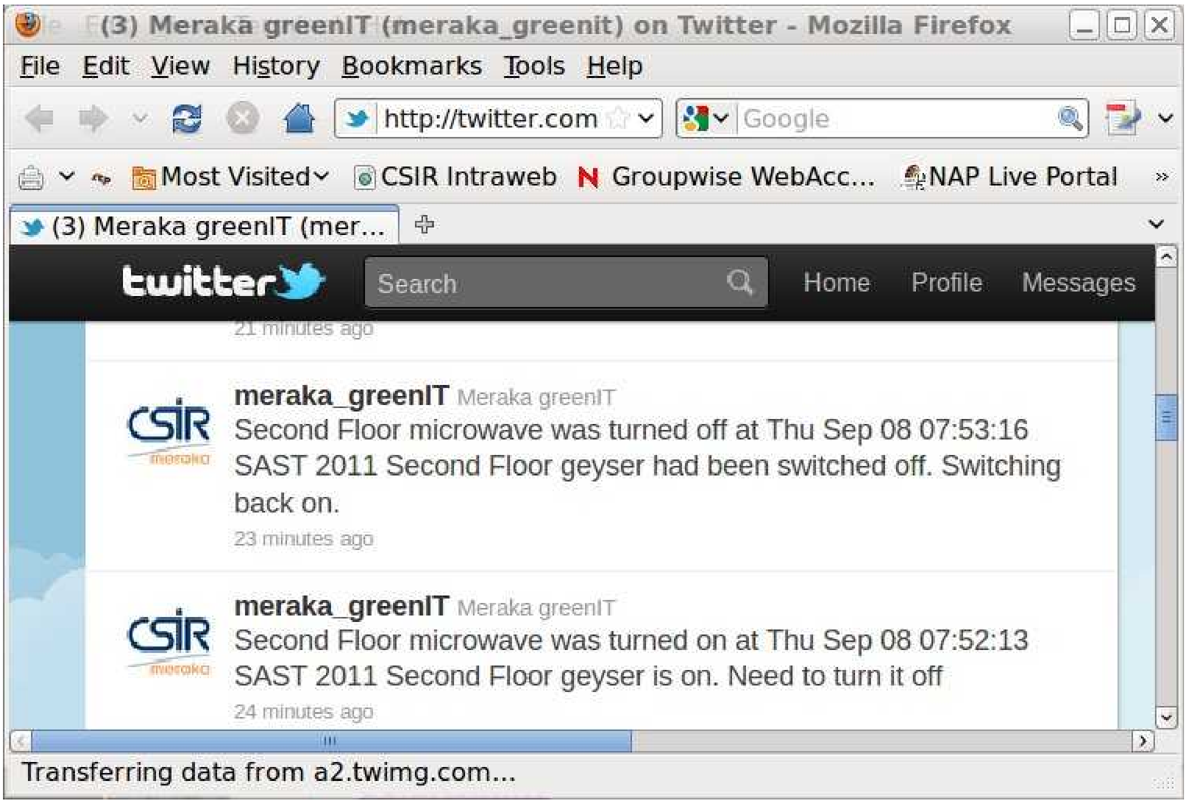


Illustration 3: Simple example of turning the geyser off then on again

Illustration 3 shows simple postings to Twitter. The messages are listed with the most recent message first. To track this scenario, the messages must be read from bottom to top. In this example, at 7:52:13 the microwave located in the second floor kitchen was turned on. The system determined that the geyser was also on at the same time. It then issued instructions for the geyser to be turned off. At 7:53:16, the microwave was turned off. The system determined that the geyser had been turned off to balance the load and then issued the appropriate instructions to turn the geyser back on.

Obviously, the system needed to receive other messages in the interim. Illustration 4 shows a more complicated example where the kettle on the second floor kitchen was turned on. The system determined that the geyser was on and should be turned off. Another appliance was turned off in the third floor kitchen and did not affect anything in the second floor kitchen. Then the second floor kettle was turned off and the system determined that the second floor geyser needed to be turned back on.

The model also catered for situation where an appliance is turned on and the geyser is, therefore, turned off. Then while the appliance is still on another appliance is also turned on. Obviously the geyser can not be turned back on until both of the smaller *on-demand* type appliances are switched off.

By following these postings on Twitter, it was possible to identify and address any major flaws with the model before implementing the actual physical switching on and off of the geyser.

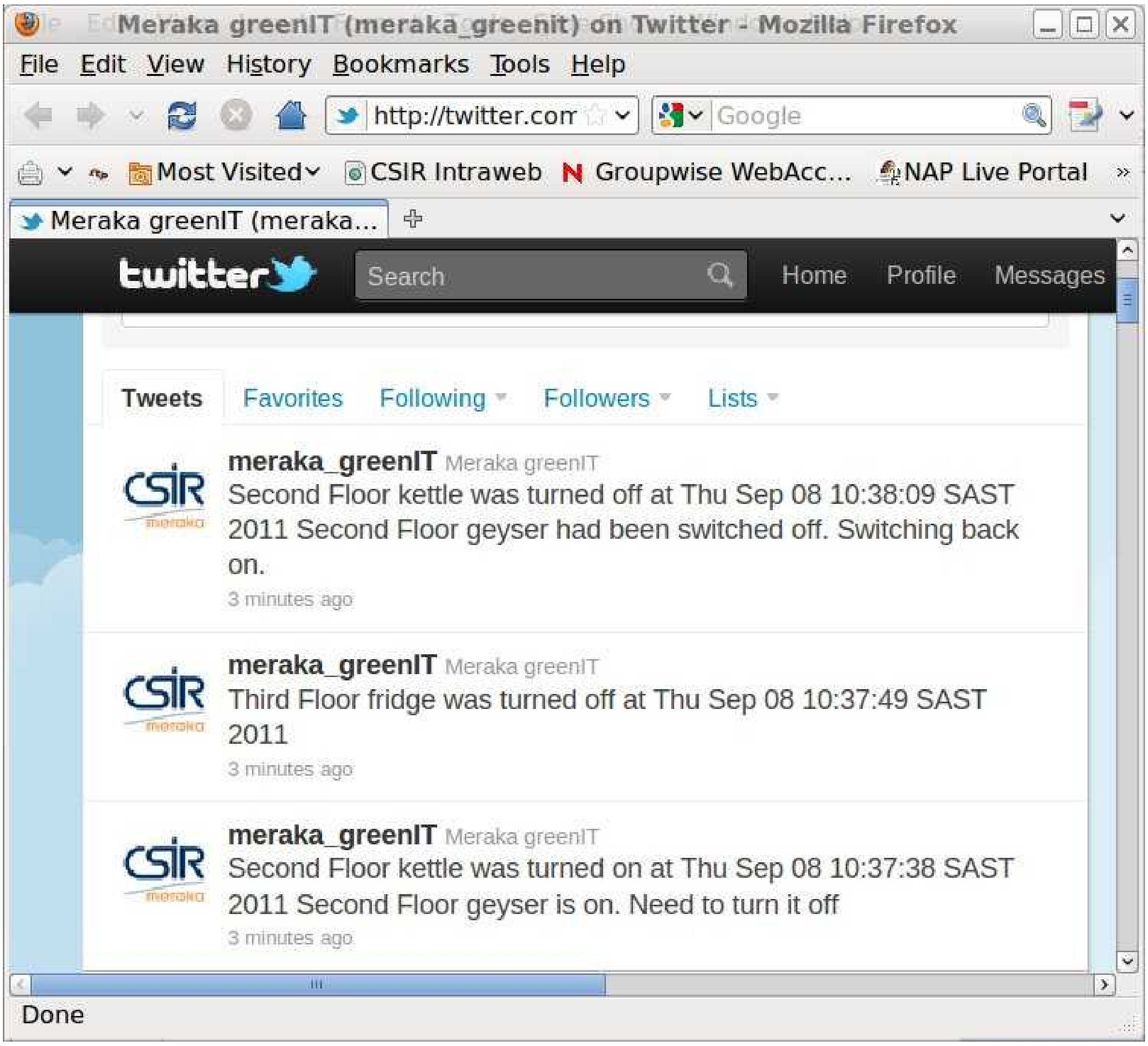


Illustration 4: Mixed messages from Current Cost

Switching Electrical Appliances On and Off

IoT (Internet of Things) appliances can be controlled by switching them on and off, or by sending them commands of varying urgency to do so. The commands are communicated to the IoT appliance via the Internet. When the command has been received, it has to be implemented in a physical form so as to have an effect in the real world. The following subsections address the categories of commands that can be sent to an IoT appliance, the mechanism used in communicating the command, and the interface to the physical world.

Command categories

We have defined four command categories to cater for various situations we expect to encounter once the system has been widely deployed. The currently defined command categories with their description and explanation are given in Table 2. Colours in the table give an indication of the urgency of the command category. Command category 3 is the most urgent and is coloured red. The colours orange, yellow, and green correspond to command categories 2, 1, and 0 respectively. Command category 0 has the lowest level of urgency and serves as a suggestion only.

Command category 3 is designed to be used when the situation is critical and the IoT appliance needs to respond immediately to the command. An example of where command category 3 is applicable is when there is a certainty that the IoT appliance will be submersed in water due to water flooding, but the appliance is not designed to be submerged under water. Were the appliance submerged, it could potentially pose an electric shock hazard. With information at its disposal, the business process can therefore make the decision to force the IoT appliance to shut down.

Command category 2 serves as an instruction to the IoT appliance to change to the requested state as soon as possible. An example of a situation where this command category could be used is when the national electricity supplier cannot provide the anticipated electricity demand. This is a preventative measure to avoid power disruptions. In this scenario an IoT domestic geyser would switch off as soon as the water being heated has reached the set temperature. The geyser would remain in the off state until a command is received, typically a category 0 command which would be a suggestion to the geyser to switch back on again.

Command category 1 has a lower level of urgency to those of categories 3 and 2. This category serves as a request for the IoT appliance to execute some action. An example of an action requested at this category is for a kettle to switch off once the water has boiled.

Command category 0 has the lowest level of urgency and such a command may be ignored. An example of this command in use is when the business process is aware that the occupants of a domestic dwelling are not at home, but the television is switched on. The command may then be sent to the IoT television to switch off.

Command category	Description	Explanation
0	A suggestion.	This command may be ignored.
1	A request.	Implement when current activity has been completed, e.g. switch the kettle off once the water has boiled.
2	Implement as soon as possible.	This command may be used when the national electricity supplier has load balancing problems.
3	Implement immediately.	This command is for emergency use only, e.g. in the case of fire or flooding.

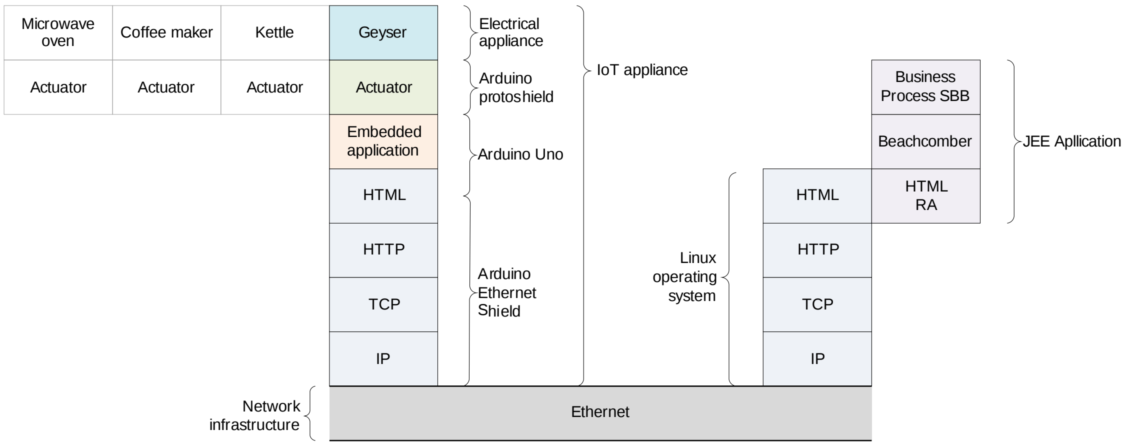
Table 2: System commands with their description and explanation

Communication

Communication between the UrbanController and Beachcomber using HTML formatted messages transiting over an HTTP/TCP/IP network stack as shown in Illustration 5.

Communication between the business process (part of the JEE application) and an appliance transverses a number of layers of a custom-developed IoT stack.

Illustration 5: The IoT communication stack



IP and TCP

At the bottom of the stack lie the IP, and TCP layers that are based on the industry standard. Due to the limited resources (both processing ability and storage) available on the Arduino Ethernet shield [10], these layers implement only the fundamental industry standard as applicable to the IP, and TCP protocols. In contrast to the IoT appliance, the JEE application is not constrained by resources allowing for robust, complex, and comprehensive implementation of the protocols.

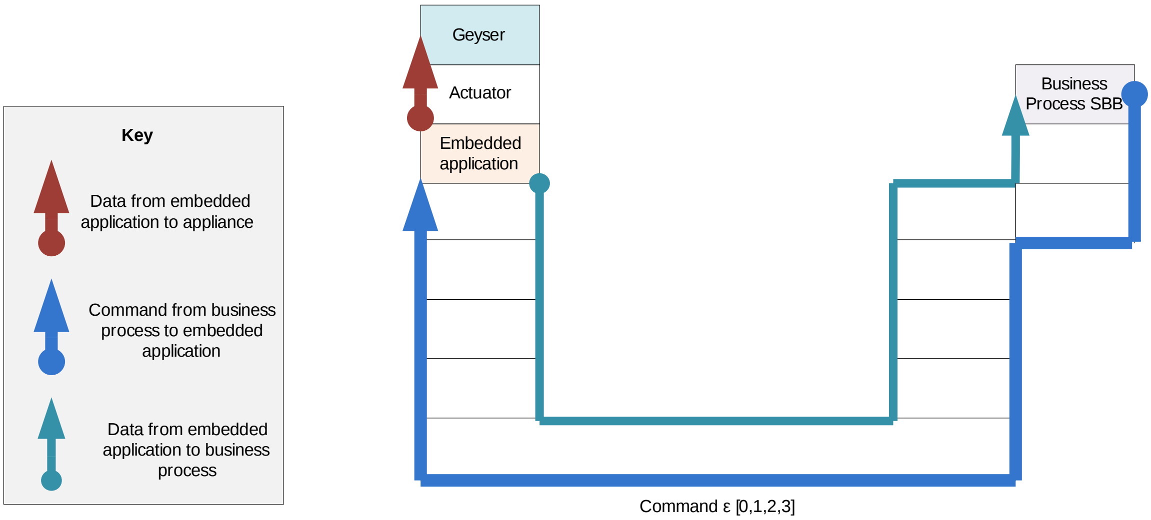
HTTP

Next in the IoT stack, the custom-developed embedded application serves as a HTTP parser and an actuator controller. This embedded application executes on an open hardware Arduino Uno [11] circuit and exchanges data with the Ethernet shield using an open-source software library. Commands received from the business process are parsed, examined, and acted upon, at the embedded application layer. A single embedded application can control multiple appliances. Information on which appliance should be controlled is retrieved at the same time as when the HTTP header is examined. Also contained in the HTTP header is the command applicable to the appliance.

HTML

The HTML layer in the stack makes provision for returning data to the business process. Data that indicate the amount of visible light and temperature in the vicinity of the IoT appliance can influence future decisions made by the business process. The data returned is in human-readable form and can be accessed by pointing a web browser at the IoT appliance’s internet address. In the current implementation, unprocessed data from the low-cost light and temperature sensors give an indication of relative level changes. These values are not calibrated but could still serve an indicators whether there are significant changes in the immediate environment.

Illustration 6: Data and communication flow between the appliance, embedded application, and the business over an IoT stack



Physical Control

When the embedded application has successfully parsed the commands sent by the business process, a decision is made which IoT appliance should be affected by either turning it on or turning it off. Our current implementation makes provision for up to five appliances to be controlled using a single Arduino Uno and Arduino protoshield circuit pair. A protoshield contains five transistor driver circuits to switch electromechanical relays. In addition to driver circuits, five red light emitting diodes (LEDs) provide a visible indication of each relay state. The relays are similar and are activated by applying 5V DC to the coil terminals. Single pole, double throw contacts can switch 16A AC. In this implementation, the normally closed (NC) contacts are wired in series with one supply lead of the appliance being controlled. This has the added advantage of returning the appliance to its normal operating state, should the electrical power to controller circuit be disrupted.

EVALUATION

Although the refrigerator was considered to be a *permanent* appliance (as opposed to an *on-demand* appliance) in this model, for the scope of this evaluation and testing, the refrigerator was not switched on or off for load balancing. This was due to potential health risks during the testing phases if the refrigerator was switched off for abnormally long periods of time.

In order to evaluate this project, the data captured by the CurrentCost equipment was stored and graphed to show the peak electrical loads as can be seen in Illustration 7. The graph is for the period of approximately 10:45 to 11:45. The key indicates that red shows the consumption of the refrigerator, orange shows the consumption of the geyser, and purple shows the consumption of the kettle.

Illustration 7 shows the results when there is no load balancing. As can be seen in the graph, at approximately 11:10 the geyser came on and started to draw electricity. At approximately 11:12 the kettle also came up. In view of the fact that there was no load balancing, the combined electrical consumption peaked to just over 3k. That is indicated by the grey value in the graph.

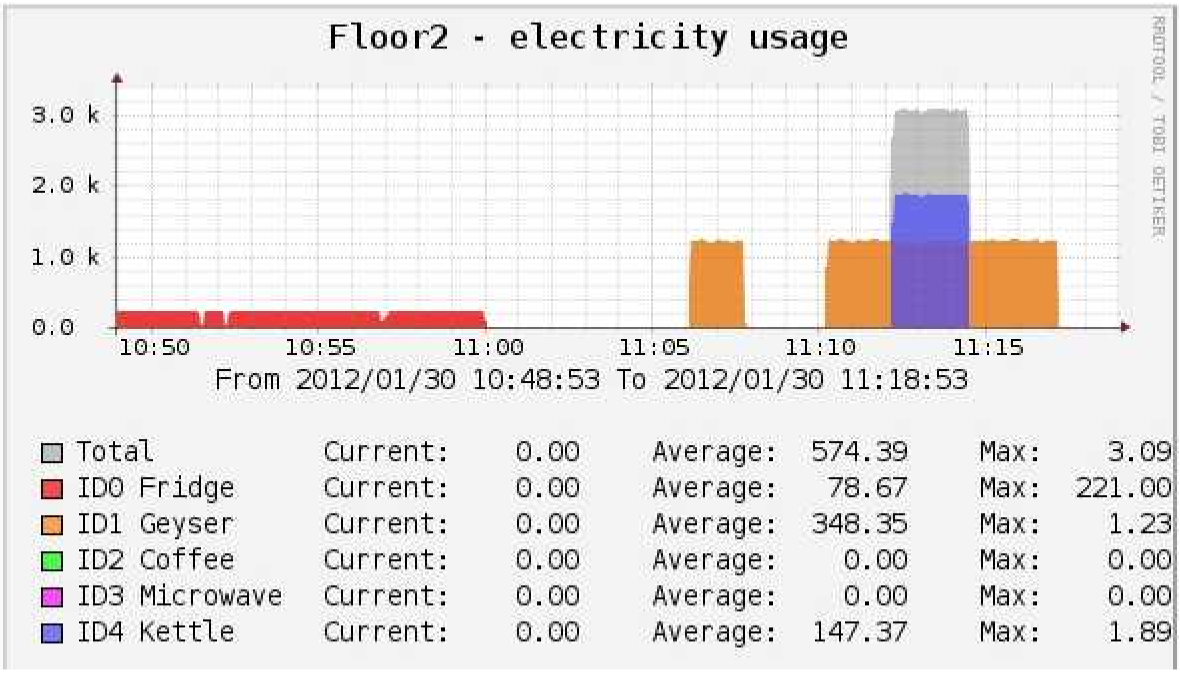


Illustration 7: Graph of electrical load, no load balancing

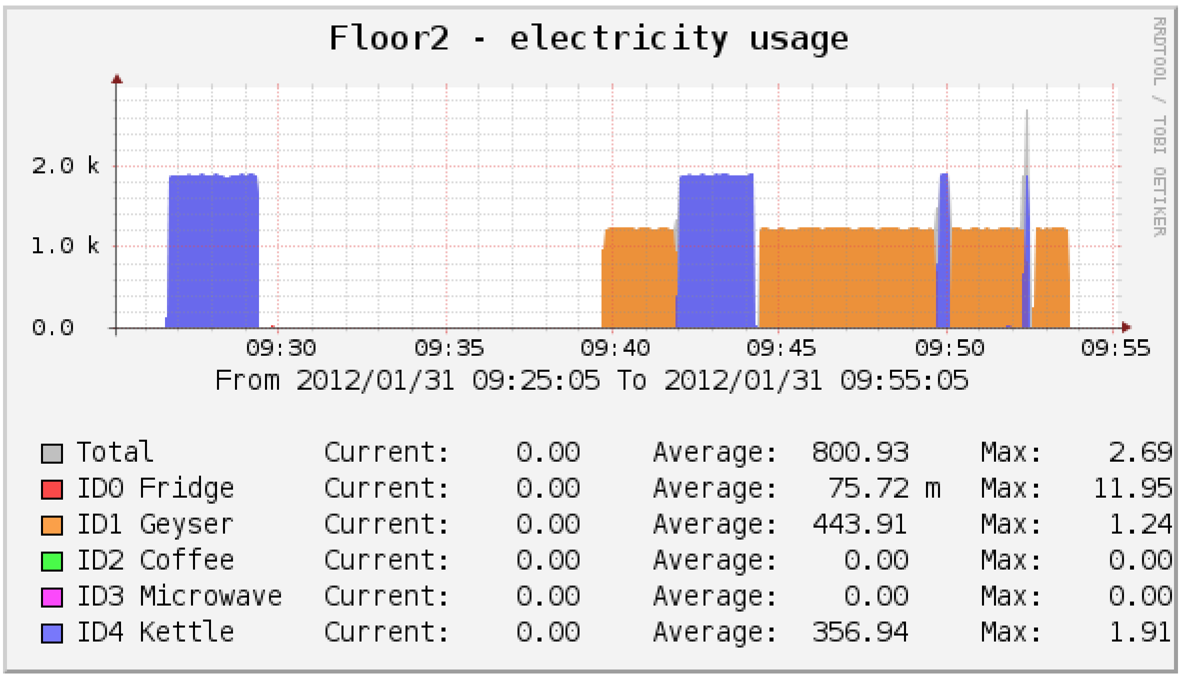


Illustration 8: Graph of electrical load, with load balancing

Illustration 8, however, shows a similar situation where the electrical load balancing is in place. At approximately 9:40 the geyser switches on. When the kettle is turned on at approximately 9:42, then the geyser is switched off. At approximately 9:45 when the kettle is switched off, then the geyser is switched back on. Although the total amount of electricity consumed is the same, the load is balanced and does not exceed 2.0k.

It is important to note that there are still occasional short spikes in the electrical consumption as can be seen at approximately 9:52. This very short spike is due to the sampling rate of the monitoring equipment. If the monitoring equipment manages to detect that the kettle has been turned on as the kettle starts up and before the kettle reaches its full consumption level, then there will be no large spike. This is illustrated on Illustration 8 at 9:42 and 9:50.

Conclusion

We have categorized and discussed the commands that may be sent to IoT-enabled appliances. It is likely that these categories may change, and the number of categories adjusted, as we gain a better understanding of the environment in which the system is deployed. We plan to close the control loop between the business process and the IoT appliance by including remotely sensed light and temperature data in the business process. Using this data, we anticipate that the business process will be able to make better informed decisions when deciding which command to send to the IoT appliance.

The experiments described in this paper confirm that it is possible to use Internet protocols and various Internet of Things techniques to balance electrical consumption. Appliances were classified as being either *permanent* appliances or *on-demand* appliances. A simple model was created in which all *permanent* appliances were switched off whenever any *on-demand* appliances were switched on. This would balance the electrical consumption.

References

[1] T. Jamasb, W. J. Nuttall and M. G. Pollitt. *Future Electricity Technologies and Systems* 2006*67.*

[2] A. Sebitosi and P. Pillay. Grappling with a half-hearted policy: The case of renewable energy and the environment in south africa. *Energy Policy 36(7),* pp. 2513-2516. 2008.

[3] A. de Saint-Exupery. Internet of things. 2009.

[4] ITU, "The internet of things, executive summary," International Telecommunications Union, Geneva, 2005.

[5] R. Bencina and M. Kaltenbrunner. The design and evolution of fiducials for the reactivision system. Presented at Proc. 3rd International Conference on Generative Systems in the Electronic Arts. 2005, .

[6] J. Caruso, "The 'Internet of Things' now includes a human heart," vol. 2010, August 10, 2009, 2009.

[7] H. Sundmaeker, P. Guillemin, P. Friess and S. Woelfflé. Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commision* 2010.

[8] B. J. Oates. *Researching Information Systems and Computing* 2006.

[9] L. Butgereit and L. Coetzee, "Beachcomber: Linking the "Internet of Things" to the "Internet of People"," *Proceedings of IST-Africa, 2011, may 11-13, Gabarone, Botswana,* 2011.

[10] M. Margolis and N. R. Weldin. *Arduino Cookbook* 2011.

[11] T. Igoe and M. Tepper. *Making Things Talk: Practical Methods for Connecting Physical Objects* 2007.