Makahiki+WattDepot: An open source software stack for next generation energy research and education

Philip M. Johnson, Yongwen Xu, Robert S. Brewer, George E. Lee, Andrea Connell

Collaborative Software Development Laboratory, University of Hawaii at Manoa, Honolulu, HI 96822 USA
johnson@hawaii.edu, yxu@hawaii.edu, rbrewer@hawaii.edu, gelee@hawaii.edu, connell4@hawaii.edu

ABSTRACT

The accelerating world-wide growth in demand for energy has led to the conceptualization of a "smart grid", where a variety of decentralized, intermittent, renewable energy sources (for example, wind, solar, and wave) would provide most or all of the power required by small-scale "micro-grids" servicing hundreds to thousands of consumers. Such a smart grid will require consumers to transition from passive to active participation in order to optimize the efficiency and effectiveness of the grid's electrical capabilities. This paper presents a software stack called comprised of two open source software systems, Makahiki and WattDepot, which together are designed to engage consumers in energy issues through a combination of education, real-time feedback, incentives, and game mechanics. We detail the novel features of Makahiki and WattDepot, along with our initial experiences using them to implement an energy challenge called the Kukui Cup.

Keywords

smart grid, energy research, open source, game engine, energy repositories

1. INTRODUCTION

Several characteristics of the traditional electrical energy infrastructure of industrial societies have remained unchanged for almost 100 years. First, energy production has been centralized in power plants using "firm" energy sources such as coal, oil, nuclear, or hydro. Second, centralized production has promoted centralized control of the grid, typically through a single or small number of utilities with public regulation over their policies and rates. Third, centralized production and control has led to the predominence of "macro-grids", or grid infrastructures designed to service hundreds of thousands to millions of consumers. Finally, traditional grids have been designed to minimize the information about energy required by consumers to utilize the service. The typical consumer needs to know almost nothing more than how to plug an appliance into an outlet, and can assume that this exceedingly simple "user interface" will provide virtually unlimited amounts of high quality energy at any time for a relatively small cost.

Unfortunately, the accelerating world-wide growth in demand for energy is leading to a breakdown in this approach to electrical energy infrastructure. Petroleum products such as coal and oil are nonrenewable, are no longer reliably inexpensive, and have been found to have a variety of adverse environmental effects. Nuclear energy, while low in emissions, has risks that have caused countries such as Japan and Germany to reevaluate this approach to energy generation.

Addressing these emergent problems has led to the conceptualization of a "smart grid", where a variety of decentralized, intermittent, renewable energy sources (for example, wind, solar, and wave) would provide most or all of the power required by small-scale "micro-grids" servicing hundreds to thousands of consumers. Such a smart grid will require consumers to transition from passive to active participation in maintaining efficient and effective use of the grid's electrical capabilities. For example, these "smart consumers" should be able to tailor their use of electrical energy to the types and amount of energy available in the grid at any point in time; minimizing the overall use of non-renewable resources as well as peak loads on the grid.

Satisfying the radically different requirements and operating assumptions of this next generation grid requires new kinds of software that enable research and experimentation into the ways that electrical energy production and consumption can be collected, analyzed, visualized, and provided to consumers in a way that enables a transition from passive to active participation. Since 2009, we have been designing, implementing, and evaluating an open source software "stack" to facilitate this research. This software stack consists of two custom systems called WattDepot[1] and Makahiki[2], along with the open source components they rely upon (Java, Restlet, Postgres, Python, Django, Memcache).

In this paper, we detail the novel features of WattDepot and Makahiki, along with our initial experiences using them to implement an energy challenge called the Kukui Cup.

2. WATTDEPOT

Software for energy collection, storage, and analysis tends to come in two flavors that support two ends of the scalability spectrum. At one end are utility-scale SCADA systems and protocols which are intended to manage macro-

ICT4S 2013: Proceedings of the First International Conference on Information and Communication Technologies for Sustainability, ETH Zurich, February 14-16, 2013. Edited by Lorenz M. Hilty, Bernard Aebischer, Göran Andersson and Wolfgang Lohmann.
http://e-collection.library.ethz.ch

[1] http://wattdepot.googlecode.com
[2] http://github.com/csdl/makahiki

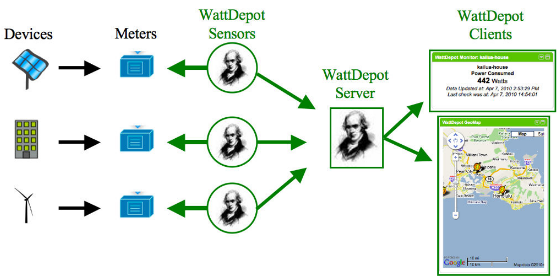


Figure 1: Architecture of WattDepot

grid data [6, 7, 9]. At the other end are "personal scale" systems such as those provided by energy meter or solar panel manufacturers which are intended to manage information about single households [4, 10]. We designed WattDepot to support a middle ground that we refer to as "enterprise-level" energy management, in which data concerning energy production and consumption of hundreds to thousands of households can be usefully managed [1]. Figure 1 illustrates the architecture of the system, where WattDepot sensors send data from meters attached to energy devices to a server, which can then be queried by clients to provide visualizations and analyses.

Our use of WattDepot has led to a novel set of capabilities to support this middle ground.

2.1 Meter Agnostic

Unlike personal-scale systems that are typically tied to a particular manufacturer's product, WattDepot is agnostic about the kinds of meters used to monitor energy production and consumption data, and whether the data is personal-scale or utility-scale. It provides a REST protocol for data transmission that can be used to implement clients for a wide variety of devices; the major constraint is that these devices need to have network access. WattDepot clients can be written in any language that supports the HTTP protocol. We provide a high-level client libraries for Java and JavaScript.

Due to the architectural decoupling of data collection from the rest of the system, WattDepot can be effectively used for simulation and what-if scenario development. This flexibility makes it appropriate as a kind of technological "scaffolding" for smart grid applications, where WattDepot can provide clients with simulated production and consumption data early in development, with the simulated data transitioning to live data as these sources go online later in development.

2.2 Source Aggregation

WattDepot can represent aggregations of power sources. For example, a building might have multiple meters monitoring energy consumption, one per floor. WattDepot can represent the power consumed by individual floors, as well as an aggregate source representing the building as a whole. Aggregations can be nested, so that floors can be aggregated into buildings, buildings into neighborhoods, and neighborhoods into cities. It is quite common for level of abstraction desired by client developers and end users (such as a floor of a building) to actually consist of multiple meters. By providing this aggregation of sources at the server level, client development becomes easier.

2.3 Data Interpolation

WattDepot automatically performs data interpolation when necessary. For example, a meter might provide a snapshot of energy usage once per hour for a given device. Clients can request the power consumed by this device at any time instant, and WattDepot will automatically provide interpolation when the requested time does not match a time for which actual sensor data is available. This is essential for the common case where meters do not have perfectly synchronized clocks and are not polled simultaneously, and when making use of the source aggregations discussed in the previous section.

2.4 Flexible Data Storage

WattDepot is architecturally decoupled from the underlying data storage technology. This supports experimentation with both traditional relational as well as NoSQL technologies, and facilitates scalability. Currently, WattDepot implements support for Derby, PostgreSQL, and BerkeleyDB storage systems. Administrators looking for simplicity may opt for the embedded Derby database, while those looking to integrate with existing database infrastructure might decide to use PostgreSQL for data storage.

WattDepot also implements support for "ephemeral" data. In some application scenarios, it is useful to send energy data to the WattDepot server quite frequently (i.e. every few seconds) so that clients can monitor current energy consumption with low latency. However, that rate of data sampling is not necessary for historical analyses, which may only require energy data sampling at the rate of every few minutes. WattDepot supports this situation through ephemeral data, which creates an in-memory window during which all recently received energy data is available for retrieval, but stored in the repository only at a much lower sampling rate.

2.5 WattDepot in the Cloud

In addition to installation on a local server, WattDepot has been designed to support cloud hosting, sometimes referred to as Platform-as-a-Service (PaaS). In particular, WattDepot can be deployed on the Heroku[3] cloud-based hosting service. The Heroku PaaS solution allows users to deploy the system and start collecting data without the requirement of server hardware, and Heroku offers flexible capacity depending on the expected workload.

2.6 Beyond WattDepot

While WattDepot provides the software infrastructure to collect, store, and analyze energy data, ensuring that the data is collected reliably and accurately reflects reality requires additional effort. As an example, we will examine the steps required to collect data on electricity use in a building. First, the administrator should work with manager of the building to understand the electrical infrastructure: how

[3] http://www.heroku.com

is power distributed in the building, and how does the distribution relate to the goals to be accomplished through measurement? If electricity is to be monitored at the per-floor level, do the distribution panels match that segmentation?

If the building does not have meters already installed, the administrator will need to select a meter vendor and acquire the meters. The meters will typically need to be installed by licensed electricians. The administrator will need to verify that the installation was performed correctly by checking whether the received data agrees with the expected amount of usage. To allow WattDepot to collect data, reliable network connectivity must be provided for the meters. The meters will also need to be configured to support remote data collection.

If the building already has meters installed, the administrator will need to confirm that the meters are configured correctly and that the data they produce is sane. When working with existing meter infrastructure, the WattDepot administrator may not have administrative access to the meters, so any configuration changes required may need to be requested from the meter administrator.

Once data collection has been established, the WattDepot administrator will need to establish monitoring of the meters and WattDepot infrastructure so that hardware or software faults can be detected and corrected before they lead to excessive loss of data. By understanding these issues, administrators can ensure that systems built on top of WattDepot receive accurate energy data at appropriate levels of abstraction.

3. MAKAHIKI

The feature set of WattDepot creates attractive infrastructure for management of energy data, but research suggests that effective participation of consumers in a next generation smart grid requires more than simple feedback to consumers about their consumption, particularly given the passive nature of their involvement for the past 100 years.

The second component of our open source software stack, Makahiki, represents research intended to create synergy between the need to create knowledge and engagement regarding energy and the ability of so-called "serious game" techniques and energy feedback to create participation and engagement [2, 3, 5, 8]. In Makahiki, online game mechanics are employed with the goal of affecting real-world energy behaviors. The ultimate goal is to not just affect energy behaviors during the course of the game, but to produce long lasting, sustained change in energy behaviors and outlooks by participants. Figure 2 illustrates the architecture of Makahiki.

Makahiki consists of a configurable game engine that can be customized to the needs of different organizations. It includes a library of pre-built game "widgets" that implement a variety of game mechanics. Using the widgets, an organization can create a custom energy challenge in which players can compete individually and/or in teams to earn the most points by reducing their energy consumption as well as by learning about energy concepts in general. The next sections present some of the most important widgets in Makahiki.

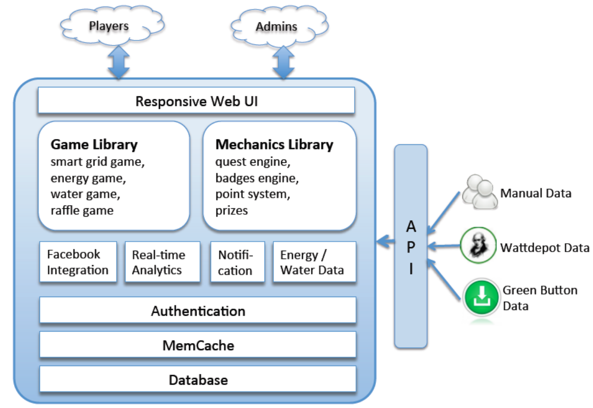


Figure 2: Architecture of Makahiki

3.1 Smart Grid Game

The *Smart Grid Game widget* shown in Figure 3, is the primary place players go to learn about energy issues and earn points. Actions are organized into a grid of squares (hence the name "Smart Grid") and organized by category columns. The grid contains four different types of actions: activities, commitments, events, and excursions.

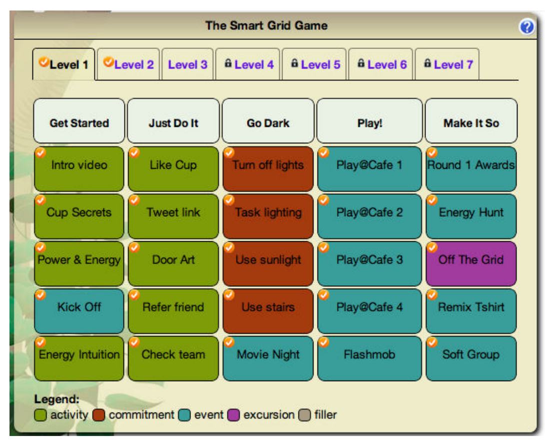


Figure 3: ***Smart Grid Game***

Activities are the most basic task available in the Smart Grid. In order to get points for an activity, a player will have to provide a response to the administrators. These responses can be a short textual answer or an uploaded picture. Administrators access a special admin section of the web application to approve or deny submissions. If a submission is approved, the player will receive the points for their submission. Otherwise, a player will be sent a website notification informing them that their submission was not approved, and a textual description by an administrator of why it was rejected. The player can change and resubmit their response and still earn the full point value for that task.

Commitments are pledges that the player will do something sustainable for a period of five days. Examples include: reducing shower time, taking the stairs, and turning off the lights when leaving a room. Because these commitments are not verifiable, they are worth fewer points than activities. Furthermore, a player can only have up to five active commitments at any given time. After the five day period is up, the player can then declare that they completed the commitment and immediately earn their points. They can then sign up for another commitment, including the one they just completed.

Events and excursions are tied to real world activities. Events are held on campus while excursions take place off campus. Seating is limited, so players are asked to sign up for events they wish to attend. Players that do so are provided with a 2 point signup bonus. Players can also set up a reminder that is sent to their email and/or their mobile phone before the event takes place. At the event, an administrator will hand out attendance codes printed on slips of paper that can be entered on the website. These attendance codes are generated by Makahiki and can only be used once. To discourage players from signing up and not attending, a 2 point penalty is assessed to players who do not submit an attendance code. If the player submits an attendance code for the event after receiving this penalty, the penalty is reversed.

Not all of the tasks in the Smart Grid Game are necessarily available at the start of the game. We implemented a set of predicates that can be used to determine if a task is locked or unlocked for a player. These predicates include: completed a certain number of tasks within a category, completed all tasks within a category, completed certain tasks, and time-based unlocking (available after a certain date).

These predicates are implemented using a limited subset of Python and can be changed within the Django admin interface. Competition designers can use logical operators to combine any of these functions in order to organize the players' path through the Smart Grid Game.

3.2 Power Meter

A fundamental requirement for enabling more active participation by consumers in the smart grid is feedback regarding their energy usage. One of the most simple mechanisms provided by Makahiki for this purpose is the Power Meter widget, illustrated in Figure 4.

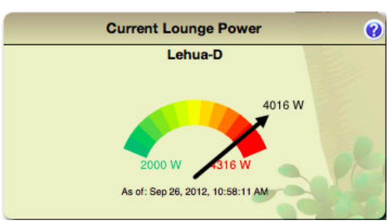


Figure 4: ***Power Meter Visualization***

The Power Meter widget provides basic feedback on energy consumption via a display of the team's power consumption, updated every few seconds. The visualization can normalized using baseline values so that when the needle is pointing straight up, the power consumption is average for that team during that specific hour of that specific day of the week. Thus, if the needle leans left toward the green side, the team's power consumption at that moment in time is below average, while if the needle leans right toward the red side, the team's power consumption at that moment in time is above average.

The Power Meter widget obtains its values by querying the WattDepot system for the latest power data consumed by the associated team. The use of WattDepot, rather than directly querying the meter(s), simplifies the widget design significantly. First, the physical meters can vary significantly in the protocol implemented to obtain current power consumption. These protocol variations are handled by the WattDepot sensors, so this widget can simply query the WattDepot server using a simple single HTTP request that is independent of the physical meter characteristics. Second, the power consumed by a team might be measured by one or multiple meters. Again, the WattDepot virtual source capability means that this physical difference can be abstracted away by WattDepot, enabling the widget to obtain the aggregate power for the team through a single HTTP request.

The Power Meter widget is a useful, though simple mechanism for energy feedback that uses the WattDepot+Makahiki stack. The next section presents a more sophisticated mechanism called the Daily Energy Goal Game.

3.3 Daily Energy Goal Game

The *Daily Energy Goal Game widget* provides a way for players to earn points by reducing their current consumption from a baseline. Both the baseline data and the current consumption is typically provided by API calls from Makahiki to an underlying WattDepot server. Figure 5 illustrates this widget.

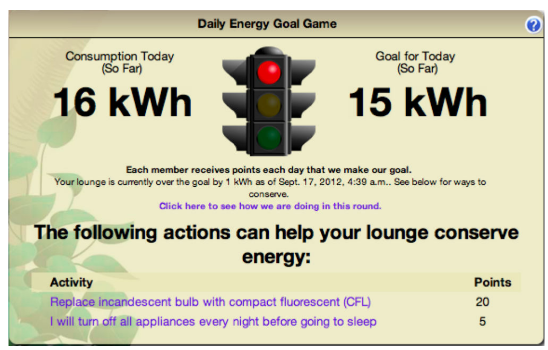


Figure 5: ***Daily Energy Goal Game***

The goal for each team is typically a percent reduction from their baseline usage of the previous two weeks. When a player goes to the energy page of Makahiki, they can view their team's current progress toward their daily energy goal. Near the end of the day, Makahiki checks the energy data from Wattdepot to see if a floor reached their goal. If the floor did reach their goal, each member of the floor that is participating in the game receives 20 points. The energy goal game provides a link between the energy conservation competition and the point competition.

The Daily Energy Goal display shows both their current progress and their goal so far for two reasons. First, everyone will be under their actual energy goal for most of the day, so this display would not be very useful. Second, we have noticed that the students in the residence halls use more energy at night rather than during the day. Thus, it is easy to be under for most of the day and then jump over the goal at the very end. Displaying their progress toward the goal so far provides a pace for players to follow.

3.4 Raffle Game

The *Raffle Game widget* provides a way to incentivize participation from all individuals, even those who are not in the running for a top prize. For every 25 points a player earns, they receive one virtual raffle ticket. Players can dynamically allocate their tickets to any raffle prizes they are interested in at any time, up to the end of the raffle. Figure 6 shows an example of the Raffle Game.

Round 2 Raffle Game					
Your total raffle tickets: 5 Allocated right now: 2 Available: 3					
Prize	**Value**	**Your tickets**	**Total tickets**	**Current odds**	**Change ticket allocation**
Recycled bike	$200.00	1	2	50.0%	+1 -1
UH t-shirt (1)	$28.00	1	1	100.0%	+1 -1
Outback card	$25.00	0	0	0.0%	+1 -1
Smart strip (2)	$25.00	0	0	0.0%	+1 -1
Smart strip (1)	$25.00	0	0	0.0%	+1 -1
Down to Earth card	$25.00	0	0	0.0%	+1 -1

Figure 6: *Raffle Game*

Players can dynamically allocate their tickets to any raffle prizes they are interested in at any time, up to the end of the raffle. Each round of the competition has its own set of raffle prizes and any unused raffle tickets carry over to the next round. Raffle tickets are independent from a player's score; allocating a raffle ticket does not affect their rank.

3.5 Social and Referral Bonuses

The *Social and Referral Bonus widgets* provide game mechanics that help encourage participation by providing additional points to players who participate in activities with other players and/or facilitate the entry of new players into an energy challenge.

The social bonus is an administrator option when a task is created in the Smart Grid Game. It awards extra points if the player has done the task with someone else. Examples of tasks with social bonus include attending an event, recording a song related to energy, or measuring a shower water flow rate. When a player submits a response for a task with a social bonus, the player can provide the email address of the person who jointly completed the task. Once the other player completes the task, the social bonus is awarded. Social bonuses are not bi-directional; if the second player doesn't provide the first player's email address, only the first player will get the social bonus.

Players are led through a setup process when logging into Makahiki for the first time. One of the steps in this process is the referral bonus. If a player was referred by another player in the system, they can use this step to input their email address. Once the new player earns 30 points in the competition, both players are awarded a referral bonus of 10 points. Typically, going through the setup process gives you 25 points, so we wanted to encourage the new player to at least complete one additional task in order to get the referral bonus.

3.6 Quest Engine

One challenge we faced when designing Makahiki was providing adequate help to the player. The game needed to be intuitive, even if a new player coming to Makahiki has not participated in an energy competition. Unlike many web applications, such as webmail, game players generally do not know in advance what specific tasks they wish to accomplish. In an effort to provide a player with guidance through Makahiki after the setup process, we implemented the Quest Engine. Quests are used to guide the player through the various workflows of the site, like completing a task, signing up for an event, or allocating a raffle ticket. These quests can be created in the admin interface. They also use a set of predicates to determine unlock and completion conditions. These predicates are: participating in a task or type of task, completed a task or type of task, has a certain number of points (in a round or overall), completed a certain number of tasks in a category or of a given type, awarded a badge, wrote a post on their floor wall, and added a picture to their profile.

3.7 Game Analytics

Makahiki is designed to support energy challenges involving hundreds or thousands of users lasting weeks or months. In these circumstances, effective use of the technology requires the ability to understand the state of the game, such as: Who is using it? What are they doing? What is the player response to activities, commitments, excursions, and events? Such state information is important for planning purposes, such as assessing the transportation needs for an upcoming excursion by seeing how many players signed up. It can also be used for making in-game changes to game analytics, such as changing the point values associated with activities to encourage or discourage participation. It can also help identify breakdowns in game play, such as significant numbers of unallocated raffle tickets indicating that users do not understand the nature of that game mechanic.

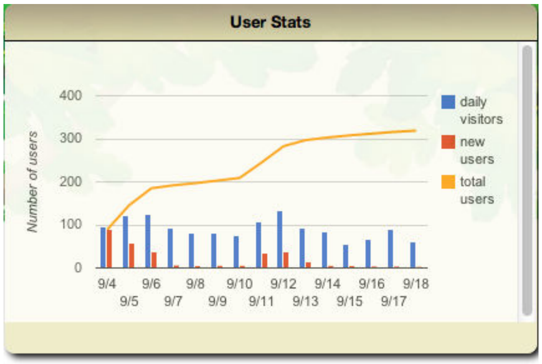
To address these needs and others, Makahiki includes a variety of widgets that work together to provide high level game play state to the administrators of a challenge. Figure 7 shows an example of two game analytic widgets.

The top widget, User Stats, shows trends in the total number of players, the total number of new users, and the total number of players visiting the site each day. The bottom widget provides information on the ability of teams to achieve their daily energy goal each day and over time.

We have now introduced the primary components of our software stack, WattDepot and Makahiki. The next section presents our experiences and lessons learned so far.

4. EXPERIENCES

As a preliminary assessment of the Makahiki+WattDepot software stack, we designed and implemented an energy chal-



Energy Goal Status

Team	09/16	09/15	09/14	09/13	09/12	09/11	09/10
Lehua-A	-16 87	-2 87	-5 86	-9 87	-8 85	-5 88	-3 87
Lehua-B	-14 101	-7 100	-14 100	-11 101	-12 100	-9 104	-13 99
Lehua-C	-9 91	-2 85	-4 90	-5 92	-4 92	-2 94	-9 92
Lehua-D	-1 94	0 91	-1 92	0 95	2 98	2 97	2 99
Lehua-E	-14 82	-6 77	-4 85	-12 88	-6 91	-4 92	-13 86
Ilima-A	-14 87	-6 89	-5 87	-13 85	-8 87	-13 86	-15 83

Figure 7: Real time game analytics

lenge called the "Kukui Cup" for over 1,000 first year students living in the residence halls at the University of Hawaii in Fall, 2011. We installed 40 Shark 200-S meters throughout the residence halls, and used the ModBus WattDepot sensors to gather instantaneous power and cumulative energy every 15 seconds.

Response to this initial challenge was very positive. Over 400 students participated, for an adoption rate of approximately 40%. In a survey of those participating students conducted near the end of the challenge, over 90% of them said they would play the game if it were offered next year. 60% of participants said "ease of use" was the thing they liked best about the website. 40% responded "Nothing" when asked what was confusing about the website, and 32% responded "Nothing" when asked what they would change about the website. The survey did yield insights into what could be improved, including the ability to introduce new games at points during the challenge, to provide better access to other player data, and simplify navigation.

5. CONCLUSIONS

Need conclusion here.

6. ACKNOWLEDGMENTS

Add acknowlegements to sponsors and other KC team members.

7. REFERENCES

[1] R. S. Brewer and P. M. Johnson. WattDepot: An open source software ecosystem for enterprise-scale energy data collection, storage, analysis, and visualization. In *Proceedings of the First International Conference on Smart Grid Communications*, Gaithersburg, MD, October 2010.

[2] S. Darby. The effectiveness of feedback on energy consumption. Technical report, Environmental Change Institute, University of Oxford, 2006.

[3] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: Defining "gamification". In *Proceedings of MindTrek*, 2011.

[4] Control4 EMS 100. `http://news.cnet.com/8301-11128_3-20000731-54.html`.

[5] A. Faruqui, S. Sergici, and A. Sharif. The impact of informational feedback on energy consumption–a survey of the experimental evidence. *Energy*, 35(4):1598–1608, 2010.

[6] Open Phasor Data Concentrator (OpenPDC). `http://openpdc.codeplex.com/`.

[7] Open Source Home Area Network (OSHAN). `http://sourceforge.net/projects/oshan/`.

[8] J. E. Petersen, V. Shunturov, K. Janda, G. Platt, and K. Weinberger. Dormitory residents reduce electricity consumption when exposed to real-time visual feedback and incentives. *International Journal of Sustainability in Higher Education*, 8(1):16–33, 2007.

[9] Smart energy 2.0. `http://www.zigbee.org/SmartEnergy`.

[10] The Energy Detective (TED) Home Energy Meter). `http://www.theenergydetective.com/`.