Exploring Power Storage Profiles for Vehicle to Grid Systems

Aaron Hunter and Ray Young

British Columbia Institute ot Technology Burnaby, BC, Canada aaron_hunter@bcit.ca, raythetse@gmail.com

Abstract

The Smart Grid allows users to monitor power usage through the use of Smart Meter technology. In principle, this information can be used to modify usage habits in a way that reduces consumer costs as well as greenhouse emissions. However, in an urban environment, many users are restricted by the same constaints: they work during the day, and they are home at night. This creates spikes in power cost at peak usage times, and it may also lead to increased emissions in scenarios where sustainable resources are limited. An individual user can avoid these spikes by using an electric car as a storage device; it can be charged at the cheapest times, and then discharged to the home at the most expensive times. While this idea is intuitively appealing, it turns out that the benefits vary greatly depending on the storage algorithm used. In this paper, we describe the Power Storage Simulator, a tool for experimenting with storage algorithms to improve the efficiency of vehicle to grid systems. We suggest that this tool is also useful for educating power consumers about load balancing on the Smart Grid through an engaging, visual simulation.

Introduction

Consumer demand for electrical power is not evenly distributed throughout the day in an urban environment. There are typically peak usage periods in the mornings and evenings, as well as very low usage periods overnight. This introduces a problem when we try to maximize the use of electricity generated from sustainable sources. At peak times, it is often necessary to draw additional power from cheaper, non-renewable sources such as coal in order to fulfill consumer needs. It has been demonstrated previously that the use of micro-storage systems can be used to balance the demand for electricity around the clock, while also reducing costs to consumers (Vytelingum et. al 2011; Voice et. al 2011). However, as smart meters are being rolled out in major urban centres, basic research and experimentation is still required to determine exactly how storage devices should be used to maximize utility. Moreover, consumers need to be educated not only about the value of storage devices, but also about the importance of intelligent power storage procedures. In this paper, we describe the Power Storage Simulator(PSS), a tool that allows users to

experiment with power storage, and view the potential benefits first hand.

PSS is built on a gaming platform; it allows developers to experiment with storage algorithms and it can also be used as an educational tool to teach consumers about the importance of power storage and load balancing. The interface is intended to be an educational game, which allows users to play with parameters and view the resulting gains. The backend is built on top of a Smart Grid ontology that permits flexible modification, and explicit specification of electrical power resources. The PSS tool makes several contributions to existing work on power management. First, it provides a platform for experimentation with different storage strategies that allows us to explore the impact of vehicle to grid (V2G) systems for load balancing. Second, it allows us to educate power consumers on load balancing for the Smart Grid through a simple simulation and visualization. At a technical level, PSS demonstrates the value of a formal ontology for representing and reasoning about electrical power.

Preliminaries

Motivation

This work was originally motivated by sustainability concerns. Many sources of "clean" energy provide a constant level of power (hydro-electricity) or intermittent power (wind). In order to meet demand at peak times using these sources, it is therefore necessary to use some form of storage. While the potential benefits of storage systems have been known for several years, the required technology has not been widely available. Suitable batteries have traditionally been prohibitively expensive, and power consumers have not been aware of the potential savings and advantages. However, as electric vehicles have become less expensive, the cost of batteries is no longer a major barrier. In this paper, we use the term "battery" to refer to a generic storage device, which in practice is likely to be an electric vehicle. Our simulation therefore focuses on one important application of V2G systems: load balancing through power storage.

PSS was originally developed as an educational game. The idea was to produce a game in which players tried to reduce greenhouse emissions by trying to employ suitable intelligent appliances. At present, the simulator is not a true game, however. Instead, the simulator is more of a test-bed

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



Figure 1: PSS Interface

with a great deal of educational content. It is a kind of *serious game*, where the incentive to explore parameters is a real-world desire to reduce costs and emissions.

Ontologies

From the outset, a secondary goal of our simulator was to demonstrate the utility of formal *ontologies* for reasoning about electrical power. An ontology is a formal specification of the elements of a domain, and the relationships between elements. One highly influential ontology language is the Web Ontology Language (OWL), and it's variants (Horrocks et. al 2003; Motik et. al 2009). OWL has many advantages when formalizing a domain in practice, so it was our goal to use OWL to formalize power resources in PSS.

The Power Storage Simulator

Overview

PSS is built in C# on Microsoft's XNA framework. The simulation consists of an electrical network that draws power from two main power generators - a hydroelectric dam, and a coal burning plant. The hydroelectric dam generates power at a low cost and without carbon emissions; the coal plant is used during peak hours when the dam cannot meet demands and produces power at a higher cost while producing carbon emissions. This creates a situation in which power stays at the same price so long as only the dam is generating power, but prices climb and emissions are introduced when the coal plant is called upon to supplement the dam's production. The simulation interface is pictured in Figure 1.

Users of the simulation will be motivated to explore ways that batteries can be used to increase the use of "clean" energy. As such, users are provided with detailed information about the amount of energy that is provided by each power source. This information is formatted in an easy-toread graphical format. One such graph for hydro-electric power use over time is provided in Figure 2.

The original intention was to use an existing ontology for the representation of power resources, e.g. (Penya et. al 2011). However, in the end, we decided to create a simple prototype based on the publicly available ontology at www.gridpedia.org. Note that our ontology is greatly simplified from the original gridpedia version, it is also distin-



Figure 2: Hydro-Electric Power Usage

guished by the fact that it is an OWL ontology created in Protege. The set of classes in our ontology is in Figure 3.

Simulation Details

The simulation runs at a pace determined by a game clock, where the power usage and emissions are updated as each unit of time elapses. We refer to each unit of time as a *span*, and it corresponds to a 30 minute duration in real time. Each update consists of the following sequence of steps:

- 1. Update the game clock.
- 2. Calculate how much power each generator can produce.
- 3. Calculate how much power each battery can discharge.
- 4. Calculate how much power each powered agent in the game will draw during the relevant period.
- 5. Draw power required from the generators and batteries.
- 6. Process all bills generated.
- 7. Aggregate all records made during the update.

Storage Algorithms

Overview

While it is known that power storage can lead to improved performance, we observed that it is also possible for batteries to "interfere" with each other in a way that actually negates the benefits. We have developed several different algorithms that can be explained to users and demonstrated. Each storage algorithm allows each update a battery to charge or discharge during a span. The basic idea is that a battery will charge when power is inexpensive and discharge when power is expensive. This means that, rather than using expensive peak-hour power, agents will have an opportunity to access cheap power generated during other times of the day. Extensive documentation is provided that would allow a software developer to make new algorithms based on user instructions.

Simple Charge or Discharge

The first algorithm implemented was really developed as a simple means for testing the system. It looks at the historical data for system-wide power usage for the current day. It then determines the 24 half-hour spans that had the lowest usage and the 24 that had the highest usage. During those spans



Figure 3: The PSS Ontology

that had the lowest usage, batteries will store power; during those spans that saw highest usage, they will discharge power as necessary. This is a commonsense algorithm that makes sense to a non-technical user, and can be used as a first demonstration.

This algorithm results in an overall reduction in power costs and emissions initially, but suffers from a major problem. All of the batteries charging at the same time creates a new peak period; this means that charging the batteries becomes more expensive than just buying power at other times. A further complication is that during the same day next week the record now shows peaks in those spans that were used to charge batteries during the previous week. The batteries then charge during very different spans. This again carries through to the next weeks, causing the peak-usage spans to "flip" and each day the batteries are drawing power during peak spans. The end result is that power costs and emissions return to where they started and even begin to grow beyond their original levels.

Weighted Charge or Discharge

The second algorithm attempts to distribute the charging of individual batteries across all low-usage spans according to the specific usage during each span. That is to say, each span is assigned a "weight" according to how much power is drawn during that span compared to the extreme spans. The algorithm is as follows:

Weighted Charge

Set A = the average value of all 48 spans for the day Set L = the value of the lowest span Set H = the value of the highest span Set V_i = the value of the current span Set R_i = the calculated ratio for power transfer If the span is lower than the average: $R_i = -1 \times (A - V_i)/(A - L)$ If the span is higher than the average: $R_i = (H - A)/(V_i - A)$

This algorithm runs into problems when one span dominates the other spans. For example if a few spans are significantly cheaper or more expensive than the other relevant spans, then the ratio value for the less extreme spans is insignificant. As a result, barely any power is charged or discharged during the insignificant spans even though it may be beneficial to do so.

Balanced Weighted Charge or Discharge

The third algorithm attempts to alleviate the problems of dominant spans observed in the previous section. Rather than simply weighting each span in an absolute manner, each span will be weighted according to the aggregate value of all "relevant" spans, where the notion of relevance is implicit in the following algorithm.

Balanced Weighted Charge

Set A = the average value of all 48 spans for the day Set L = the value of the lowest span Set H = the value of the highest span Set NL = the number of below average values Set NH = the number of above average values Set V_i = the value of the current span Set R_i = the calculated ratio for power transfer Calculate the total of the difference: Total = 0For each span value V_i If $V_i < A$ $Total + = V_i - A$ Calculate weight of each span: If span value > average $R_i = (V_i - A)/Total$ If span value > average $R_i = (A - V_i)/Total$

Note that, in this algorithm, the rates produced no longer refer to the fraction of the maximum transfer rate of the battery but a fraction of the total capacity of the battery. This algorithm proved to be more effective than the previous algorithms and greatly reduced the problems associated with dominant spans. The algorithm still suffers from "flipping" however.

Balanced Weighted Charge or Discharge with Interpolated Graduation

The final algorithm tested also uses the Balanced Weighted Charge to calculate the ideal charge and discharge rates, given current information about power usage. However, instead of immediately setting transfer rates to match the ideal, we gradually interpolate from the last results towards the upto-date results. By making the change gradual, we never swing beyond the ideal rate. In this manner, we effectively avoid the flipping problem.

The algorithm is an extension of the Balanced Weighted Charge from the previous section. Each battery starts with an array of 48 values representing the charge/discharge rate for each span that day. Each day, the array for that day will be updated by moving some fraction of the way towards the newly calculated ideal. The battery uses the newly calculated ideals to determine how much it wants to charge/discharge for each span:

- Set *C* = the maximum capacity of the battery.
- Set R_i = the rate calculated as before for a particular span.
- Set *L* = the array of charge/discharge values for the particular battery (all values initialized to 0).
- Set L_i = the local rate for the battery and a particular span.
- Set F = the fraction of interpolation (defaults to 0.1).

Using these values, we calculate:

$$L_i = L_i + F * (R_i * C - L_i).$$

This is by far the most effective approach tested to date. It proved very efficient and displayed non-volatile responses to changes in the system. This approach continued to show benefit when batteries were permitted to move. This is clearly an important property for V2G systems, but it is a property that was not demonstrated by the previous algorithms.

Discussion

Extensibility

The original intention was to use an ontology of power resources in order to produce software that was elaboration tolerant. However, the use of our ontology has been limited due to performance constraints with our current 1/480s refresh rate. Moreover, in many cases the class of power consumers and power generators will actually change over time. As such, we suggest that our internal data representation could be improved through the use of efficient ontology evolution operators. Some aspects of this problem are addressed in (Magka et. al 2013), but the general problem of how to efficiently perform ontology evolution is still open.

Algorithm Specification

Our goal in the development of this software is twofold. First, we would like to produce a tool that can be of use to planners and developers experimenting with different algorithms for power storage. To a great extent, this is possible in the current version of the software. The second goal was to create an educational tool that could be used to teach the general population about the potential advantages of intelligent power storage. This goal has only been partially accomplished. The current version of the simulation provides users with two main capabilities. Specifically, users can specify whether or not the homes monitor energy usage with intelligent appliances and users can put batteries in homes for storage. By manipulating these parameters and viewing the extensive information displays, the user can learn a great deal about the way power storage improves performance. However, the user is not currently able to *change* the storage algorithm from within the simulation. In order to specify a storage algorithm, the user needs to be able to specify a procedure through a simple interface for non-programmers. This is a challenging problem in interface design that we leave for future work.

Conclusion

We have described the development and operation of PSS: a tool for simulating power storage on the Smart Grid. Our software demonstrates the viability of power redistribution across all hours of the day through the use of intelligent storage and discharge systems. Moreover, we have demonstrated that the system can be used to experiment with different storage algorithms. Our preliminary results show that some algorithms reduce emissions and costs, while other algorithms do not. The most effective algorithm presented shows definite improvement over a storage-free system, and avoids the problem of demand synchronization.

While we have focused on the analysis of different storage algorithms in this paper, the simulation is really intended to be a kind of serious game. We are interested in educating regular power users on the advantages gained through power storage. We are also interested in demonstrating to these same users that these advantages are only achieved if we define suitably intelligent algorithms for controlling power distribution.

References

Horrocks, I., Patel-Schneider, P., and van Harmelen, F. 2003. From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Journal of Web Semantics* 1(1):7-26.

Magka, D., Krtzsch, M., and Horrocks, I. 2013. Computing Stable Models for Nonmonotonic Existential Rules. In Proceedings of IJCAI 2013.

Motik, B., Patel-Schneider, P., and Parsia, B. 2009. OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. W3C Recommendation. http://www.w3.org/TR/owl2-syntax/.

Penya, Y., Garbajosa, J., Ortega, M., and Gonzalez, E. 2011. ENERGOS: Integral smart grid management. In *Proceedings of the IEEE International Conference on Industrial Informatics*.

Voice, T., Vytelingum, P., Ramchurn, S., Rogers, A. and Jennings, N. Thomas Voice, Perukrishnen Vytelingum, Sarvapali D. Ramchurn, Alex Rogers, Nicholas R. Jennings 2011. Decentralised Control of Micro-Storage in the Smart Grid In *Proceedings of AAAI-2011*.

Vytelingum, P., Voice, T., Ramchurn, S., Rogers, A. and Jennings, N. 2011. Theoretical and practical foundations of large-scale agent-based micro-storage in the smart grid. *Journal of Artificial Intelligence Research*, 42, 765-813.