Context-Aware and User-Centric Residential Energy Management

Baris Aksanli  
Department of Electrical and Computer Engineering  
San Diego State University  
San Diego, CA  
Email: baksanli@sdsu.edu

Jagannathan Venkatesh¹, Christine Chan²,  
Alper Sinan Akyurek², Tajana Simunic Rosing¹  
¹Department of Computer Science and Engineering  
²Department of Electrical and Computer Engineering  
University of California San Diego  
La Jolla, CA  
Email: {jvenkate, csc019, aakyurek, tajana}@ucsd.edu

Abstract—The Internet of Things (IoT) has brought increased sensing, monitoring and actuation capabilities to several domains including residential buildings. Residential energy management methods can leverage these capabilities and devise smarter solutions. This requires processing and reasoning data constantly generated by various IoT devices. In this paper, we use a hierarchical system model for IoT-based residential energy management, that includes a general purpose functional unit to drive data processing and reasoning. We apply this hierarchy to represent the electricity delivery structure from the utilities to individual residences. Our system captures additional data generated by various devices as user context and uses this context to determine user flexibility towards energy management. Our experiments show that modeling user context brings over 14% improvement in energy flexibility prediction accuracy and 12% reduction in annual grid energy cost.

I. INTRODUCTION

Residential energy automation has gained significant attention over the last years due to potential savings applied over millions of houses. Previous studies leverage reschedulable appliances, energy storage devices and renewable energy sources (e.g. solar, wind) to increase the energy efficiency of residential domain. Although actual instrumentation is very important to demonstrate the real applicability of the automation solutions, high-level models are also essential for testing different scenarios and scaling up the results to several hundreds/thousands of houses before real deployment [1]. Most existing studies use static models and over-optimistic improvements - using the additionally available data as ground truth. However, this data is 1) not always readily available, and 2) strongly tied to the behavior and preferences of users [2] [3]. The advent of the Internet of Things (IoT)—a collection of sensors and actuators backed by the existing and growing Internet infrastructure [4]—can provide the context needed for user-driven residential energy automation. In this work, we aim to use this additional user context to determine individual appliance and house energy flexibility. Our approach includes modeling, training, and generating this context information in an efficient and accessible way.

Pre-IoT work in ubiquitous sensing still envisioned a level of compatibility and control over the sensors in the systems [5] and applications that used a manageable amount of raw sensor data. The number of available sensing and actuation devices has grown rapidly in the last few years [6], promising a truly pervasive sensing and actuating environment. In addition, ubiquitous connectivity and cloud storage have largely mitigated the primary research issues in the pervasive sensing fields. Reliability of communication and storage allows us to focus on the application layer: IoT applications operate in a world of changing inputs and available compute nodes as sensors and devices enter and exit an application’s domain. The raw data in these applications will go through several levels of abstraction, combination, or distillation to produce a high-level description of the environment (and its users) with discrete, semantic states called context. Discrete, high-level context facilitates intuitive reasoning at the cost of raw data precision, and can be reused across applications.

These context-aware Internet of Things applications are ideally suited for determining user behavior for the residential smart grid: their main goal is to leverage the available data to drive automated actuation, such as in smart environments, distributed microgrids, or user-centric automation. They operate on dynamically changing, ontologically-defined data called context data whose type, range, and sources are specified in an interface. Current context-aware applications are still end-to-end implementations tightly coupled to the initial infrastructure and platforms, where each application maintains its own data and user interactions. As the number and heterogeneity of sensing devices and compute nodes available to each application changes, these implementations adapt poorly without complete redesign. Smaller, simpler functional units that provide intermediate steps towards an overall application can alleviate scalability issues. Additionally, the state of the art [7] [8] places the burden of processing in black-box applications. This is particularly inefficient when multiple applications need to process the same data using similar computation (e.g. both workplace automation and home security can infer a user’s location and occupancy from various data sources in the same way). Furthermore, reliance on application-specific code squanders the potential for designing and reusing general-purpose machine learning for multiple context-aware applications.

This paper identifies a novel approach to context-aware res-
idential energy management problem using a general-purpose functional unit (context engine), which drives data processing for a given output context variable. We apply this idea towards the residential smart grid as a distributed, hierarchical context-aware application. We use heterogeneous data from different residences and user activities and scale the system up with more individual compute nodes and grid elements, demonstrating the potential for complexity and overhead reduction. We then demonstrate the impact of the addition of user context, with over 14% improvement in energy flexibility prediction accuracy and 12% reduction in annual grid energy cost.

II. RELATED WORK

There is a large body of work focusing on residential energy management. These studies include data-set based studies [9], [10], [11] [12], [13], [14], [15], physical space instrumentation [16], [17] and simulation-based studies [1]. The dataset-based studies leverage time of use datasets from different countries to make a connection between daily activities of the users and the energy consumption of different appliances. Example data sets include American Time Use Survey (ATUS) [18] and Residential Energy Consumption Survey (RECS) [19]. Kolter et al. [16] and Barker et al.[17], on the other hand, instrument some houses to collect this information from specific spaces. Although this idea creates the perfect link between the users and the energy consumption, the applicability is highly limited due to scalability issues of instrumentation. Simulation-based studies, e.g. [1], solve the scalability problem but they face with validation and verification issues since the models used in simulation studies should be verified to make sure that they represent the actual living environment.

Most studies employing residential energy management have a big assumption: reschedulable appliances. With this assumption, the appliance usage instances can be treated as computer workloads and shifted to time intervals with lower electricity price to minimize the total energy cost. When rescheduling these appliances, it is important to account for user satisfaction. Previous studies account for this by having deadlines for the reschedulable appliances. Previous studies use predetermined, fix deadlines to represent these deadlines [1], [20], [21]. Other studies [22], [23] determine these deadlines randomly, but this process cannot accurately represent the actual human constraints. In this paper, we show that such fixed assumptions on user requirements can lead to deviations from actual user requirements, and thus negatively affect the potential savings of residential energy management solutions.

III. CONTEXT-AWARE SYSTEM MODELING

This section shows our system design to automate residential energy using context aware modeling. We first briefly overview our context engine architecture to organize complex systems as modular, functional units and then present how we set up multiple context engines to control residential energy.

A. Context Engine Framework

We first briefly introduce our context engine design, which designs and implements an alternate view of IoT applications: a hierarchy of multiple-input-single-output (MISO) functional units called context engines to improve reasoning and scalability while reducing the data redundancy across applications, and accomplishing the same functionality as the previous monolithic multi-input multi-output (MIMO) units. In exposing intermediate data, we reduce the complexity and improve the scalability of other applications in the larger infrastructure. The improvement in scalability may come at the cost of accuracy, but we both quantify the additional error and illustrate how it can be minimized. We exploit the unique opportunity in IoT where reasoning and data is often replicated between different applications. Modularization generates intermediate context that can be shared among applications. Furthermore, as the smaller, hierarchical functional units represent a simpler data translation compared to the overall computation of an application, we can implement a general machine learning algorithm to perform data transformation - from the input context to the output - and reduce application-specific code.

IoT applications consume data about both physical and virtual system entities. This data, from heterogeneous sources including sensors, social media, and even manually submitted by users is raw and noisy requires processing by applications to be filtered and distilled into usable information. Additionally, from the input data, applications need to extract context: high-level abstracted data. In the IoT, context tends to be human-centric classifications (e.g. location, activity) that are important to many different applications [4]. Black-box implementations of applications from raw data to output mask both types of processing output (preprocessing and common intermediate context) from other applications, which leads to redundancy in computation. Our proposal of a hierarchy of functional units in place of monolithic implementations trades off compactness for versatility. A hierarchical approach breaks down a single application into multiple functional units, increasing organizational complexity. Although serializing the process can increase latency if a highly compact algorithm was expanded, it can also expose intermediate output for reuse by other applications, thus reducing compute redundancy in the system. We will prove that it also decreases overall compute complexity and enables system scalability, in terms of reduced input processing and reduced functional order when certain conditions are met. Additionally, splitting single-step appli-
Flexible Interval

the

predictions back to our initial study, using HomeSim [1], using
back to the utility for processing. Finally, we connect our
it with the current state-of-the-art: sending all the raw data
We demonstrate the feasibility of this approach by contrasting
houses, a neighborhood, and ultimately, the residential sector.
ated context to determine the energy flexibility of a group of
loads). Furthermore, as the smart grid is naturally distributed,
interval (potential energy savings in kWh by shutting down
utility requires: energy prediction and the flexibility of the next
be passed in directly to the utility, which in turn would use
a redesigned application to provide energy prediction. This
represents a significant increase in both communication and
processing overhead. However, the context engine approach
can be used to provide only the high-level context that the
utility requires: energy prediction and the flexibility of the next
interval (potential energy savings in kWh by shutting down
loads). Furthermore, as the smart grid is naturally distributed,
we can further break down data aggregation along the existing
lines of power distribution: waystations, junction boxes, and
substations, which already have some level of computation
ability (see Figure 2). The result is a multi-tier context-aware
application that uses available residential data to determine the
flexibility of the loads of a house, and further uses this gener-
atied context to determine the energy flexibility of a group of
houses, a neighborhood, and ultimately, the residential sector.
We demonstrate the feasibility of this approach by contrasting
it with the current state-of-the-art: sending all the raw data
back to the utility for processing. Finally, we connect our
predictions back to our initial study, using HomeSim [1], using
the Flexible Interval context engine to provide reschedulable
time frames for different appliances and the Energy Prediction
context engine to generate individual appliance traces. We then
simulate the houses to quantify the cost savings of appliance
flexibility, taking into account more realistic, personalized
deadlines and exploiting variable time-of-use pricing.

B. Context Engine Setup

In our approach, we begin at the level of each individual
end-use appliance in a house. Some appliances are less flexible
(e.g. HVAC systems, refrigerators, and always-on devices)
than others whose energy is dominated by direct user in-
teraction (kitchen and laundry appliances, lighting, etc.). We
exploit the advent of smart appliances with onboard embedded
systems as potential nodes of computation. The goal in this
stage is to identify 1) user interaction with the appliance, if
applicable, and 2) whether this usage is flexible at a given
time. These intermediate outputs are further used to predict
the energy usage of the appliance in the next interval, and
consequently, the predicted energy flexibility. The intermediate
and final outputs are trained with ground truth as following:

- User interaction and activities are boolean values derived
  from analyzing the energy and/or water traces to find how
  appliances are used.
- Binary energy flexibility for appliances is derived from
  the distribution of use over time (see Figure 3. This is
  unique to each house due to differences in user behavior.

These first-stage context engines’ outputs are further used to
predict the appliance usage. While the energy usage alone
was previously used in time-series to predict future intervals’
output, we can better learn the profiles of user-triggered
appliances by leveraging user context. An individual house can
aggregate its flexibility prediction, passing it up to the next tier:
junction boxes or substations, which in turn feed aggregated
flexibility prediction to the utility. While aggregated flexibility
is useful for identifying the energy that can be saved, our
approach takes the next step and determines the individual
loads that combine to provide this flexibility. This granularity
is an innovation enabling the smart grid to perform automated
residential demand-response: feedback control signals to au-
tomate individual loads in a scalable manner.

C. Input/Intermediate Data

Our data is sourced from the Pecan Street database [25], a
dataset that aggregates individual energy and water loads. In
addition, a subset of houses provides basic information about
the number and type of occupants. We selected and replicated
houses that fall into one of the house types in Table I to
represent a neighborhood with disparate amounts and types of
data. The first-stage context engines need to be trained with
ground truth for user interaction and binary flexibility of each
interval. As Pecan Street does not provide this information
directly, we define flexibility based on historical data about
the appliances - each house show different usage patterns for
each appliance, with each cluster having a range of start times.
For each new appliance event, we assume its flexibility to
meet that of the historical operation of that appliance instance.

Fig. 2. A context engine approach to residential energy management,
with individual homes providing higher-level context in place of raw data,
aggregated and passed . The outputs per house can vary depending on the
types of sensors and actuators available to each unit.

Our data is sourced from the Pecan Street database [25], a
dataset that aggregates individual energy and water loads. In
addition, a subset of houses provides basic information about
the number and type of occupants. We selected and replicated
houses that fall into one of the house types in Table I to
represent a neighborhood with disparate amounts and types of
data. The first-stage context engines need to be trained with
ground truth for user interaction and binary flexibility of each
interval. As Pecan Street does not provide this information
directly, we define flexibility based on historical data about
the appliances - each house show different usage patterns for
each appliance, with each cluster having a range of start times.
For each new appliance event, we assume its flexibility to
meet that of the historical operation of that appliance instance.

Our data is sourced from the Pecan Street database [25], a
dataset that aggregates individual energy and water loads. In
addition, a subset of houses provides basic information about
the number and type of occupants. We selected and replicated
houses that fall into one of the house types in Table I to
represent a neighborhood with disparate amounts and types of
data. The first-stage context engines need to be trained with
ground truth for user interaction and binary flexibility of each
interval. As Pecan Street does not provide this information
directly, we define flexibility based on historical data about
the appliances - each house show different usage patterns for
each appliance, with each cluster having a range of start times.
For each new appliance event, we assume its flexibility to
meet that of the historical operation of that appliance instance.
TABLE I
THE FOUR DIFFERENT HOUSE TYPES RETRIEVED FOR THE CASE STUDY, WITH THEIR CONSTITUENT COMPONENTS.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. The aggregated instances of washing machine usage on Mondays in House B, illustrating 3 clusters of varying flexibility.

For example, Figure 3 illustrates the usage pattern of washing machines for House B, highlighting the aggregate number of instances at each time interval. The resulting clusters identify the windows of flexibility. A future appliance interval occurring at noon will have a flexibility range between 9:30 AM and 1:00 PM. Similarly, we record clusters for all appliances in all tested houses, generating unique, heterogeneous flexibility ranges that represent different user preferences. Similarly, we associate flexibility of usage to other appliances based on related research and the traces themselves. For example, the electric vehicle has three states: not plugged in, plugged in but not charging (nominal drain from charging circuit), and charging. The second and third states represent a flexible time frame for flexible use. Table II highlights the other flexible appliances. Finally, to calculate grid energy cost, we use time-of-use pricing obtained from independent system operators across the United States: CAISO for California [28], NEISO for New England [29], and ERCOT for Texas [30]. This revealed the benefits of our prediction across different pricing schemes: Boston (high mean price, high standard deviation), San Diego (medium mean price, medium standard deviation), and Houston (low mean price, low standard deviation).

TABLE II
APPLIANCE FLEXIBILITY PARAMETERS

<table>
<thead>
<tr>
<th>Flexible Appliance</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clothes Washer/Dryer</td>
<td>Flexible usage patterns [26]</td>
</tr>
<tr>
<td>Dishwasher</td>
<td>Flexible usage patterns [27]</td>
</tr>
<tr>
<td>Electric Vehicle</td>
<td>Observed flexible charging in Pecan Street dataset</td>
</tr>
<tr>
<td>Lighting</td>
<td>Variation in light intensity [1]</td>
</tr>
</tbody>
</table>

Fig. 4. Wholesale electricity prices scaled up to retail residential pricing.

TABLE III
AVERAGE MEAN ABSOLUTE ERROR (MAE) FOR EACH CONTEXT ENGINE IN SINGLE-STAGE AND SEQUENTIAL APPROACHES

<table>
<thead>
<tr>
<th>Node Type (complexity)</th>
<th>Single-stage MAE (%)</th>
<th>Avg. Context Engine MAE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (3rd order)</td>
<td>27.15</td>
<td>27.15</td>
</tr>
<tr>
<td>B (3rd order)</td>
<td>14.23</td>
<td>14.23</td>
</tr>
<tr>
<td>C (3rd order)</td>
<td>9.81</td>
<td>9.81</td>
</tr>
<tr>
<td>D (3rd order)</td>
<td>6.16</td>
<td>6.16</td>
</tr>
<tr>
<td>Single-stage (3rd order)</td>
<td>26.94</td>
<td>14.34</td>
</tr>
<tr>
<td>Context Engine Aggregator (1st order)</td>
<td>14.34</td>
<td></td>
</tr>
</tbody>
</table>

IV. GRID ENERGY SAVINGS

We expand upon our previous work [1] by connecting this user activity and flexibility prediction. Our approach integrates
HomeSim, introduced by [1], with the context engine approach to manage residential grid energy consumption and cost. In our previous work, we quantify the ability for smart appliances to be rescheduled through awareness of time-of-use retail energy prices [1]. While we previously used static flexibility information for each appliance (i.e. a fixed threshold for each appliance instance), we now have the ability to generate individual flexibility predictions using context engines (see Figure 2). To restate, we use historical appliance start times to generate a flexibility range for each appliance (see Table II). We use HomeSim’s reschedulable appliance scheduler, described in [1], which allows us to use our predicted flexibility and energy consumption to rescheduling flexible appliances. The goal is to move appliances to intervals where they can be more cheaply utilized. We use the outputs of our context engines, the predictions of energy consumption and flexibility, to generate new schedules for flexible appliances. We compare the results against having full knowledge of the appliance’s consumption data and the assumed flexibility that was determined by historical usage.

Figure 5 illustrates the three cases we investigate:

- **Static schedule**: assumes a fixed schedule for each appliance. This is derived from Table IV, which is provided by our previous work [1]. This case does not take into account differences derived from historical appliance traces. For example, the washer and dryer flexibility ranges are 12 hours and dishwasher flexibility range is 6 hours. These ranges can also be seen as deadlines, i.e. the appliance usage instances should be accommodated before the end of the flexibility interval.

- **Oracle schedule**: uses the ground truth derived from historical usage as the flexibility interval for each appliance. This varies from appliance to appliance and from house to house. Since we are using the actual energy and flexibility traces, we also have the benefit of foresight: determining the full range of the flexibility interval before and after the actual appliance instance’s start time.

- **Predicted schedule**: This is the real-time schedule determined solely by the output of context engine predictions. In addition to predicted flexibility, since our context engine only predicts one interval in advance, we only have the ability to use the remaining intervals of the flexibility range after the predicted appliance start time.

To generate the energy and flexibility predictions for each appliance, we obtain the output of the Flexible Interval and Energy Prediction context engines (Figure 2), respectively, for all subsequent intervals of the current day and pass this data into HomeSim. We assume knowledge of 24-hour time-of-use (TOU) pricing in order to facilitate rescheduling. This is data that is typically available on the wholesale sector from various ISOs [28] [29] [30]. As retail energy integrates TOU pricing as well, we expect to see similar forecasts.

Table V has three schedules (as described above) and we calculate the savings for each schedule individually as the difference between the base cost (row 1) and the cost with reschedulable appliances (row 2) of the respective schedule. We obtain 12% annual cost savings for the rescheduled appliances using individually predicted flexibility values for each appliance. We further compare this to having oracle knowledge of all appliances and their flexibilities, the ground truth that we used earlier to train the context engines, which generates 14% cost savings. The static schedule presented in the previous section (Table IV) demonstrates a further 4% savings, at 18% electricity cost saved.

Our predicted schedule’s energy consumption is within 89% and 96% of the oracle scheduler for the base energy cost and rescheduled appliance energy cost respectively, and there is only a 2% difference between our savings and that of the oracle schedule. The error in energy cost is partly due to energy prediction error, and partly because upon predicting an appliance’s start time, we only have until the rest of the flexibility interval from the predicted start time to schedule the appliance (the red interval in Figure 5). The oracle, however, has a priori knowledge of the day’s schedule, and can reschedule an appliance anytime within the flexible interval, even before the original start time (the green interval in Figure 5). Finally, the original static flexibility case study (see Table IV) yields a further cost reduction primarily because of the increased range of flexible intervals (the blue interval in Figure 5). The static flexibility interval for the clothes washer is

![Figure 5](image-url)
12 hours (Table IV), but the flexibility interval generated by historical appliance use was shown to be 9.25 hours, with a median interval of 7.25 hours. This difference, observed over all flexible appliances, provides heterogeneity in user preferences at the expense of fewer opportunities to try to save on electricity cost. Comparing the static flexibility interval to the oracle’s, we find that although it improves cost reduction by 29% (18% savings over 14% of the oracle scheduler), 10% of the statically rescheduled appliances actually fall outside their flexibility interval, missing users’ perceived flexibility deadlines. In other words, the static scheduler overestimates user flexibility and thus leads to appliance deadline violations.

V. CONCLUSION

Electricity delivery systems are being equipped with smart devices (e.g. sensors, actuators, etc.) at all levels (e.g. junction boxes, individual homes, etc.) as part of the IoT. In this work, we propose a hierarchical, modular and context-aware system architecture to leverage these smart devices in order to manage residential energy. This approach significantly reduces computational overhead, performing 96x faster than a traditional black-box application, while sacrificing only 14% accuracy on average. We further use this architecture to model user flexibility when rescheduling various appliances and exploit varying time-of-use energy prices while maintaining user satisfaction. Current static appliance schedulers consistently overestimate the degree of freedom in terms of user flexibility, thus missing real deadlines. Our predictive scheduler uses multiple context engines to energy consumption and flexible intervals of each appliance. By intelligently managing user context, we obtain a 12% reduction in annual grid energy cost (with only 2% difference in savings), while meeting all appliance deadlines.

ACKNOWLEDGMENT

This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA; and ARPANET Optimized Distributed Energy Systems (NODES) DE-FOA-0001289.

REFERENCES


